

TARTU ÜLIKOOL

Loodus- ja täppisteaduste valdkond

Tehnoloogiainstituut

Antti Jaaniste

**Robotmannekeenide haldustarkvara ning selle juurde kuuluvate
komponentide parandamine ja edasiarendamine**

Magistritöö (30 EAP)
Arvutitehnika eriala

Juhendajad:

Helina Kitsing
Karl Kruusamäe

Tartu 2017

Resümee / Abstract

Robotmannekeenide haldustarkvara ning selle juurde kuuluvate komponentide parandamine ja edasiarendamine

Käesolev lõputöö keskendub mannekeenide haldustarkvara ja sellega seotud komponentide parandamisele ja arendamisele. Töö eesmärgiks on luua komplektne tarkvara, mis koondab kogu funktsionaalsuse, mida mannekeenide haldamise juures on vaja.

Töös antakse ülevaade robotmannekeenidest ning tarkvara seisust enne töö alustamist. Viimasele tuginedes määratakse nõuded uuele haldustarkvarale. Enne tarkvara arendamist uuritakse, millised on erinevad tarkvara testimise meetodid ning kuidas robotite vigu ennustada ja ennetada. Pärast tarkvaraarendamise lõppemist uuritakse, kuidas saab aktuaatorite voolutarbimise infot kasutada vigade leidmiseks ja ennetamiseks.

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia, T125 Automatiseerimine, robotika, control engineering

Võtmesõnad: robotmannekeen, haldustarkvara

Development of management software and connected components for robotic mannequin

The topic of this thesis is development of management software and connected components for robotic mannequin. The main goal is to develop a software which has all the necessary functionality for mannequin management.

This thesis will give an overview about robotic mannequins and in which state the software was before. Software requirements are derived from the missing functionality of the previous software version. Before software development, testing and fault preventment methods are reviewed. After that there is a research how to use actuator current consumption info to find and prevent errors.

CERCS: T120 Systems engineering, computer technology, T125 Automation, robotics, control engineering

Keywords: robotic mannequin, management software

Sisukord

Resümee	2
Joonised	5
Tabelid	6
Lühendid, konstandid, mõisted	7
Sissejuhatus	9
1 Robotmannekeeni ülevaade	11
2 Olemasoleva tarkvara ülevaade	13
2.1 Aktuaatori ja silla püsivara	14
2.2 Robotmannekeeni haldustarkvara	14
2.3 Mannekeeni rakendusliides	15
2.4 Veebirakendus <i>Support Software</i>	16
2.5 Andmebaasi API	16
2.6 Tarkvara <i>Test Unit</i>	16
3 Nõuded tarkvarale	17
3.1 Vajadused	17
3.2 Funktsionaalsed nõuded	17
3.2.1 Arendajale olulised nõuded	17
3.2.2 Tehnilisele toele olulised nõuded	18
3.2.3 Ühine osa	18
3.3 Mittefunktsionaalsed nõuded	19
3.4 Olemasoleva haldustarkvara probleemid	20
3.5 Kasutusjuhud	21
3.5.1 Aktuaatori konfiguratsiooniväärtuste kustutamine (puhastamine) .	21
3.5.2 Aktuaatori konfiguratsiooniväärtuste uuendamine	21
3.5.3 Aktuaatori püsivara uuendamine	22
3.6 Kasutajarollid ja õigused	23
4 Eeltöö ja uuringud	25
4.1 Intuiitiivne kasutajaliides	25
4.2 Tarkvara testimine	26
4.3 Vigade ennetamine ja ennustamine	27
4.3.1 Ülevaade	27
4.3.2 Vigade ennustamine mannekeenil	28

5	Tarkvaraarendus	30
5.1	Mannekeenide haldustarkvara	31
5.1.1	Kasutaja sisselogimine	31
5.1.2	Sakk <i>Diagnostics</i>	31
5.1.3	Sakk <i>Testing</i>	33
5.1.4	Sakk <i>Blanking</i>	34
5.1.5	Sakk <i>Node Installation</i>	35
5.1.6	Sakk <i>Configuration Update</i>	36
5.1.7	Sakk <i>Firmware Update</i>	37
5.1.8	Sakk <i>Recycle</i>	38
5.1.9	Parameetrite kasutamine haldustarkvara käivitamisel	38
5.2	Mannekeeni tarkvara rakendusliides	39
5.3	<i>Support Software</i> 'i rakendusliides	40
5.4	Veebirakendus <i>Support Software</i>	40
5.5	Aktuaatori püsivara	40
6	Aktuaatori voolutarbimise analüüs	42
7	Robotmannekeenide haldustarkvara testimise plaan	44
7.1	Üldtest	44
7.2	Kasutajarollid ja sisselogimine	44
7.3	Sakk <i>Diagnostics</i>	45
7.4	Sakk <i>Testing</i>	45
7.5	Sakk <i>Blanking</i>	45
7.6	Sakk <i>Node installation</i>	46
7.7	Sakk <i>Configuration update</i>	46
7.8	Sakk <i>Firmware update</i>	46
8	Tulemused ja edasine arendus	47
	Kokkuvõte	48
	Summary	49
	Kirjandus	50
	Lisad	52
	Lisa 1. Kasutajarollide õigused haldustarkvaras	52
	Lisa 2. Aktuaatorite keskmised voolutarbed	53
	Lisa 3. Aktuaatori mehaanika vead	55
	Lihtlitsents	56

Joonised

1.1	Mannekeeni kasutamine pildistamisprotsessis. [4,5]	11
1.2	Aktuaatorite tüübid. Kõige vasakpoolsem on pikkusega 55 mm ning kõige parempoolsem pikkusega 165 mm.	12
2.1	Tarkvarakomponentide omavahelised seosed	13
3.1	Aktuaatori ühendamine <i>blankeriga</i>	21
3.2	Aktuaatori konfiguratsiooniväärtuste kustutamine.	22
3.3	Aktuaatori konfiguratsiooniväärtuste uuendamine	22
4.1	Aktuaatori voolugraafikute eeldatav välimus vigade korral. Graafikul on kujutatud ideaalset olukorda, kõverat ja kahjustatud keermelatti või puru keermes ning üldist viga, mis põhjustab suurenenud voolutarvet.	29
5.1	Tarkvarakomponentide omavahelised seosed	30
5.2	Sakk <i>Diagnostics</i>	32
5.3	Sakk <i>Testing</i>	33
5.4	Aktuaatorite valimine voolugraafikute kuvamiseks	34
5.5	Sakk <i>Blanking</i>	35
5.6	Sakk <i>Node Installation</i>	35
5.7	Sakk <i>Configuration Update</i>	36
5.8	Sakk <i>Firmware Update</i>	37
5.9	Sakk <i>Recycle</i>	39
6.1	Aktuaatori 21 voolutarve (punane) ning terve hammasrattaga aktuaatori voolutarve (sinine)	43
6.2	Aktuaatori 84 voolutarve (punane) ning terve keermelatiga aktuaatori voolutarve (sinine)	43
8.1	Mannekeeni V25.1 aktuaatorite keskmise voolutarbe histogramm	54
8.2	Terve hammasratas (vasakul) ja aktuaatori 21 kahjustatud hammasratas (paremal)	55
8.3	Terve keermelatt (vasakul) ja aktuaatori 84 kahjustusega keermelatt (paremal)	55

Tabelid

2.1	Aktuaatorite info kujutamine haldustarkvaras	15
8.1	Erinevate kasutajarollide õigused	52
8.2	Mannekeeni V25.1 aktuaatorite poolt tarbitavad voolud. Rohelisega on märgitud voolutarve 40 ± 10 mA ning punasega sellest suurem voolutarve.	53

Lühendid, konstandid, mõisted

aktuaator - robotmannekeenil kasutatav lineaartäitur, mis võimaldab mannekeeni kehakuju muuta.

aktuaatori identifikaator (ID) - määrab aktuaatori asukoha mannekeenil ning seda väärtust kasutatakse käskude saatmiseks konkreetsele aktuaatorile.

aktuaatori konfiguratsiooniväärtused - aktuaatori parameetrid, mida saab lihtsasti muuta. Nendeks on käigupiirang, aktuaatori pikkus, identifikaator ning keermelati tüüp.

aktuaatori koordinaat - näitab, kui palju on aktuaator välja liikunud. Näiteks, kui aktuaator on koordinaadil 10 mm, siis see tähendab, et aktuaator on algsest asendist 10 mm pikem.

aktuaatori käigupiirang - tarkvaraliselt piiratud maksimaalne pikkus, milleni aktuaator liikuda võib.

blanker - tööriist, mis võimaldab aktuaatori konfiguratsiooni algväärtustada.

CAN-siin (*Controller Area Network*) - mikrokontrollerite jadasiinvõrk, mis ühendab omavahel reaalarakendustega töötavaid seadmeid, sensoreid ja ajameid. [21]

EEPROM (*Electrically Erasable Programmable Read-Only Memory*) - elektriliselt ümberprogrammeeritav püsimälu. Kasutatakse põhiliselt mikroprotsessorites väikese hulga andmete hoidmiseks, mis ei kustu, kui toide välja lülitada. [1]

hex-fail - failiformaat, kus binaarinfot hoitakse teksti kujul. Põhiliselt kasutatakse seda formaati mikrokontrollerite püsivara ning EEPROM-i programmeerimiseks. [9]

HTTPS (*Hypertext Transfer Protocol over SSL*) - hüpertexti edastusprotokoll, kus kogu andmevahetus kasutaja ja veebiserveri vahel on krüpteeritud. [21]

JSON (*JavaScript Object Notation*) - andmeedastusformaad andmeobjektide saatmiseks, mis koosneb omadus-väärtus paaridest. [8]

KDE (*K Desktop Environment*) - rahvusvaheline vaba tarkvara kommuun, mis tegeleb vabavara arendamisega. [20]

mannekeeni asend - mannekeeni kehakuju, mis vastab kindlatele etteantud mõõtudele.

mannekeeni lukk - sillaga ühendatud elektriline lukk, mis on mannekeenile lisatud täiendava turvameetmena. Kui võtit ei ole, siis mannekeeniga ei saa teha ühtegi toimingut, mis muudaks aktuaatori püsivara või EEPROM-is olevaid väärtuseid.

Luku avamisel on lubatud eelnevalt kirjeldatud operatsioonid ning lisaks toimub ohurežiimi ignoreerimine. Luku võtit omavad ainult need isikud, kes tohivad mannekeeni juures teha mannekeeni jaoks ohtlikke operatsioone, näiteks aktuaatori püsivara või EEPROM-is olevate väärtuste muutmine.

mannekeeni ohurežiim - mannekeeni olek, kus aktuaatorite liigutamine on peatatud, sest silla või aktuaatori püsivara on tuvastanud vea.

mini-mannekeen - tarkvaraliselt komplektne mannekeen, mis koosneb ainult kontrollplaadist ning mille külge saab ühendada ainult ühe aktuaatori.

PTD (*Programming Tool for Developers*) - mannekeenide haldustarkvara, mis võimaldab arendajatel mannekeenide staatust vaadata ning aktuaatorite püsivara ja konfiguratsiooni uuendada.

R2 funktsionaaltest - aktuaatorite elektroonikaplaatide testimise süsteem. Lisaks testimisele võimaldab see mugavalt aktuaatorite püsivara vahetada.

rakendusliides - reeglistik olemasoleva valmisprogrammiga suhtlemiseks. [21]

sild - mannekeeni juhtplaat, mis sisaldab roboti versiooni infot ning ühendab omavahel aktuaatorid ning arvuti.

Support Software - veebirakendus mannekeenide versioonide ja konfiguratsiooni-väärtuste haldamiseks.

teek - kogumik kompileeritud komponente, mille funktsioone saab tarkvaras korduvkasutada. [21]

UID (*Unique Identifier*) - unikaalne identifikaator [7]. Seerianumber, mis määrab üheselt aktuaatori.

Sissejuhatus

Uute rõivaesemete ostmisel võtab erinevate suuruste selgaproovimine tihtipeale palju aega ning enne sobiva rõivasuuruse leidmist võib tekkida tüdimus. Veel raskem on sobivat suurust valida veebipoest riide ostmisel, sest siis puudub võimalus ise rõivaid selga proovida. Ei ole võimalik ka näha, kuidas rõiva lõige kehakujuga sobib. Selle probleemi lahendamiseks on loodud robotmannekeenid, mis võimaldavad näidata, kuidas kindla suurusega riideese erinevate kehakujude seljas välja paistab. Mannekeene pildistatakse erinevates suurustes rõivaesemetega, et katta võimalikult palju erinevaid kehakujusid. Rõivastega mannekeenide pilte kasutatakse veebipoodides, et ostjal oleks lihtsam leida endale sobivaid rõivaid.

Robotmannekeeni elutsüklil jaguneb kolmeks: mannekeeni arendus, mannekeeni kasutamine rõivaste pildistamisel ning aegunud disaini ja ehitusega mannekeeni kasutuselt eemaldamine. Mannekeene on valmistatud tänaseks üle kolmekümne ning igasse eelnimetatud kategooriasse kuulub vähemalt üks mannekeen. Robotmannekeene täiendatakse arendamise käigus pidevalt ning pildistamises kasutatavaid mannekeene on vaja regulaarselt hooldada. Täiendusi tehakse elektroonikas, mehaanikas ja püsivaras. Iga tehtud muudatust on vaja testida ning kontrollida, et kogu funktsionaalsus töötab ka pärast parandamist samamoodi nagu varem. Parandusi tehes on vaja kontrollida, kas viga on eemaldatud ning ülejäänud osad on töökorras.

Mannekeenidele tehtud muudatuste ja paranduste ajalugu on vaja säilitada, et näha kas üht kindlat tüüpi vigu esineb rohkem kui teisi. See on üheks sisendiks arendusele, andes tagasisidet selle kohta, mida on vaja mannekeeni disainis ja ehituses muuta.

Vaja on tarkvara, mis võimaldab mugavalt teostada mannekeenide elutsükli käigus vajalikke toiminguid. Selle kasutajad jagunevad kaheks: arendajad ning tehniline tugi. Arendajad kasutavad rakendust põhiliselt kuni mannekeeni kasutusele võtmiseni pildistamise töös ning tehniline tugi hakkab tarkvara kasutama pärast seda. Tehnilise toe ülesanneteks on mannekeenidele regulaarsete kontrollide tegemine, et avastada kasutamise käigus tekkivaid vigu ning nende parandamine.

Antud magistritöö eesmärgiks on luua rakendus, mis abistab mannekeeni arendajaid ja tehnilist tuge mannekeeni elutsükli kõigis etappides. Töö käigus määratakse nõuded tarkvarale. Töö lähtepunktiks on mannekeeni jaoks varasemalt loodud tarkvarakomponendid ning olemasolev esialgne versioon robotmannekeeni haldustarkvarast. Töö käigus muudetakse olemasolevat tarkvara ning täiendatakse seda vajaminevate funktsioonidega. Loodav tarkvara ei ole eraldiseisev osa, vaid on seotud paljude teiste tarkvarakomponentidega. Kaks suuremat komponenti, mida haldustarkvara oma tööks vajab, on andmebaasiga ühendumine ning mannekeeni juhtimine.

Töö tulemuseks valmib uus haldustarkvara versioon, mis sisaldab nõuetes kirjeldatud funktsionaalsust. Lisaks on välja töötatud testid, mille abil saab hinnata vastavust nõuetele ning avastada edasisel arendamisel kõrvalekaldeid soovitud käitumisest. Enne tarkvaraarendust on vaja uurida, kuidas sarnastes rakendustes vigu ennetatakse ning pärast tarkvara valmimist on vaja uurida kas ja kuidas saab aktuaatorite voolutarbimise infot vigade ennustamiseks ja ennetamiseks kasutada.

1 Robotmannekeeni ülevaade

Robotmannekeen on kuju muutev mannekeen, mida kasutatakse Fits.me veebipoodide jaoks loodud virtuaalse proovikabiini teenuses. Mannekeenile pannakse rõivas selga ning ühte riidesuurst pildistatakse erinevates mannekeeni asendites. Iga mannekeeni asend vastab erinevale etteantud mõõtudega kehakujule. Klient saab rõivakaupluse veebilehel sisestada enda mõõdud ning näha, kuidas veebipoest väljavalitud rõivaese tema kehakujule sobib ning selga istub.

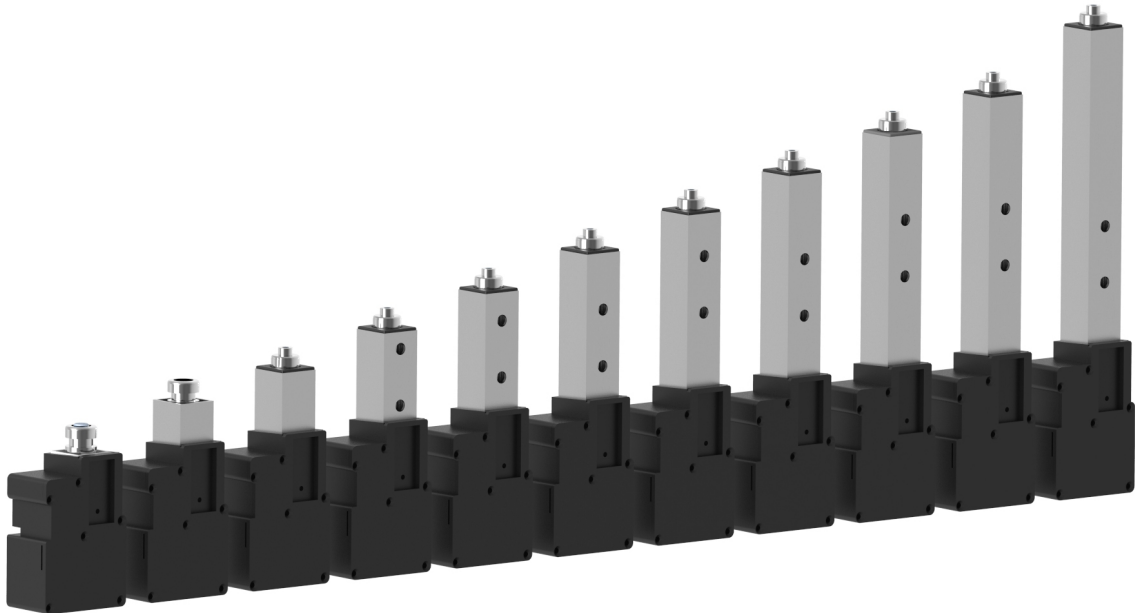
Pildistamisel juhitakse mannekeeni arvutist. Virtuaalse proovikabiini teenuse jaoks on igale mannekeenile seadistatud 50-100 kehakuju asendit. Pildistamises kasutavate asendite valik on määratud mannekeeni selga pandud rõiva omadustega nagu riide venivus ja lõige. Pildistamise tööprotsess on iteratiivne ning igal iteratsioonil saadetakse mannekeenile käsk kehakuju asendi sisse võtmiseks. Mannekeen liigub määratud asendisse ning mannekeenist tehakse pilt. Pildistamisprotsessi illustreerib joonis 1.1.



Joonis 1.1: Mannekeeni kasutamine pildistamisprotsessis. [4,5]

Mannekeeni suuruse muutmiseks kasutatakse lineaaraktuaatoreid. Need on ühendatud CAN-siinile koos sillaga. Viimane on omakorda ühendatud arvutiga üle USB-ühenduse. Mannekeeni kuju saab muuta saates käsked üle USB ühenduse või saates pakette otse

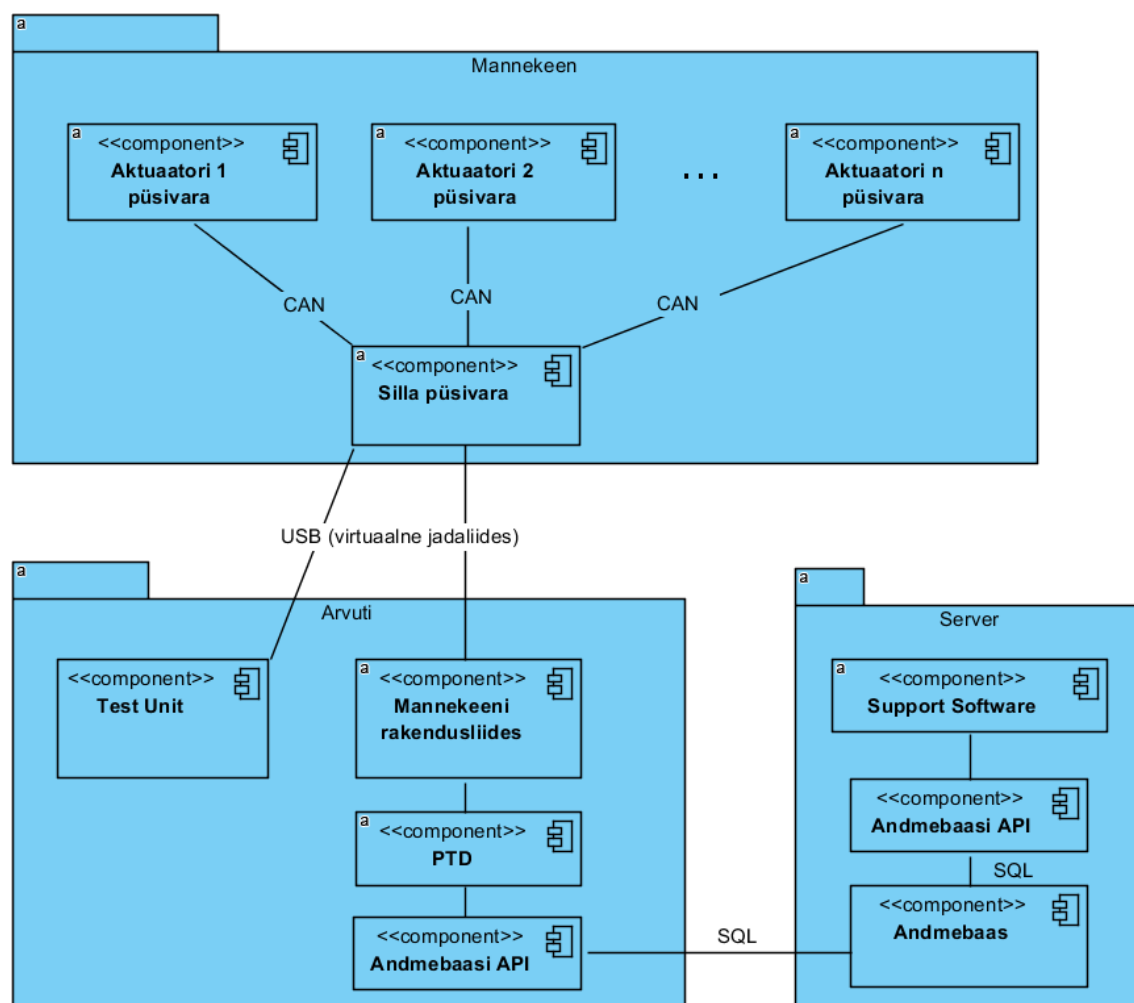
CAN-siinile. Viimast võimalust kasutatakse ainult testimise ja silumise eesmärgil. Igal robotmannekeenil on sõltuvat mudelist kasutusel 50-55 lineaaraktuaatorit. Aktuaatorite tüübid on standardiseeritud. Neid on üheteistkümne erineva pikkusega, vahemikus 55-165 mm. Iga pikkusega aktuaatorit on kahte tüüpi: 2,5 mm ja 5 mm sammuga keermelatiga. Erineva pikkusega aktuaatorid on joonisel 1.2.



Joonis 1.2: Aktuaatorite tüübid. Kõige vasakpoolsem on pikkusega 55 mm ning kõige parempoolsem pikkusega 165 mm.

2 Olemasoleva tarkvara ülevaade

Järgnevalt on välja toodud olemasoleva robotmannekeeni haldustarkvara ehk PTD seis. See tarkvaraversioon on üheks käesoleva töö lähtepunktiks. Kuna PTD ei ole eraldiseisev tarkvarakomponent, siis on kirjeldatud ka komponendid, mida PTD kasutab ning mis sellest sõltuvad. Joonisel 2.1 on kujutatud tarkvarakomponentide omavahelised seosed.



Joonis 2.1: Tarkvarakomponentide omavahelised seosed

Mannekeenil on kaks osa, aktuaatorid ning sild, mis kasutavad erinevat püsivara. Need kasutavad omavaheliseks suhtluseks CAN-pakette. Mannekeen on arvutiga ühendatud USB kaabliga ning andmevahetus toimub üle virtuaalse jadaliidese. Haldustarkvara

ning *Support Software* kasutavad andmebaasiga ühendumiseks andmebaasi API-t, mis omakorda ühendub otse andmebaasiga.

Andmebaasis hoitakse mannekeenide ja aktuaatorite infot ning logi ja veainfot. Iga mannekeeni kohta on kirjas selle üldised parameetrid, näiteks versiooninumber ja sugu ning millist tüüpi aktuaatoreid mannekeen kasutab. Iga mannekeeni aktuaatori kohta on kirjas selle identifikaator ja käigupiirang. Füüsiliste aktuaatorite kohta hoitakse infot eraldi. Kirjas on UID, kas ja millise mannekeeniga aktuaatorid ühendatud on, tüübiinfo: pikkus ja keermelati tüüp ning püsivara versioon.

2.1 Aktuaatori ja silla püsivara

Aktuaatori püsivara juhib selle mootorit ning kuulab CAN-siini pealt tulevaid pakette. Olulisemad nendest on liikumise pakett, mis ütleb millisele koordinaadile liikuma peab (ehk kui pikaks aktuaatori minema peab) ning püsivara ja konfiguratsiooniväärtuste uuendamise paketid. Lisaks saadab aktuaator CAN-siinile perioodiliselt staatuse infot, kus on kirjas kas ja millises veaolukorras antud aktuaator on. Koostamisel programmeeritakse kõik aktuaatorid sama püsivara versiooniga. Aktuaatori tüüp määratakse pärast selle koostamist, kirjutades aktuaatori EEPROM-i konfiguratsiooni väärtused.

Aktuaatori käigupiirang määrab, kui palju aktuaator välja liikuda võib. Aktuaatoritel ei saa kasutusolukorras alati lubada välja liikuda nii palju kui need mehaaniliselt võimelised on, sest pikemad aktuaatorid võivad kinni kiiluda liiga pika jõu õla korral. Teiseks kaitsevad piirangud mannekeeni mehaanikat. Aktuaatoreid ei saa liigutada välja rohkem, kui mannekeeni mehaaniline ehitus lubab. Vastasel juhul võib mõni detail ära painduda või murduda. Kolmandaks on piirangud seatud selliselt, et mannekeen saaks sisse võtta suurima kehakuju, mida tellija on soovinud.

Silla püsivaras on defineeritud mannekeeni üldinfo. Selles on määratud mannekeeni sugu, mudeli number ning millise identifikaatoriga ja millist tüüpi aktuaatorid peavad olema ühendatud. Lisaks saadab silla püsivara CAN-siinile tulevaid pakette arvutisse ning vastupidi. Sild kontrollib ka aktuaatorite staatust. Veaolukorra tuvastamisel saadetakse vastavasisuline info arvutisse ning mannekeen läheb ohurežiimi, kus aktuaatoreid enam liigutada ei saa. See on vajalik, et vältida mannekeeni võimalikku kahjustumist aktuaatori tõrke tõttu.

2.2 Robotmannekeeni haldustarkvara

Sõltuvalt konkreetsest mudelist, kasutatakse ühes mannekeenis 14-16 erinevat tüüpi täiturit. *Support Software*'i andmebaasis on iga mannekeeni versiooni kohta määratud, millist tüüpi ning milliste konfiguratsiooniväärtustega aktuaatorid mannekeenis kasutusel on. Roboti koostamisel on vaja aktuaatorite konfiguratsiooni muuta, sest pärast aktuaatorite koostamist on kõigil sama ID ning tüübile vastavad vaikimisi konfiguratsiooniväärtused.

PTD üheks põhiliseks ülesandeks on aktuaatorite seadistamine vastavalt mannekeeni

udelile ning füüsilisele kohale selles mannekeenis. Aktuaatoreid on vaja seadistada nii uue mannekeeni koostamisel kui ka hoolduste käigus asendamisel. Lisaks on seda vaja käigupiirangute uuendamisel. Igale aktuaatorile määratakse käigupiirang mannekeeni projekteerimisel ning see on ka esialgne väärtus, mis aktuaatorisse kirjutatakse. Arendusjärgse testimise käigus võib ilmned, et mannekeeni kehakujus ei ole mõned piirkonnad piisavalt toetatud, mistõttu võib aktuaator kinni kiiluda ning selle asukohta on vaja muuta. Mannekeeni mehaanika disaini muutes ei pruugi algselt määratud käigupiirangud enam sobivad olla ning neid on vaja uuendada. Lisaks sellele on vahepeal vaja piiranguid suurendada, kui tellija soovib mannekeenile rohkem liikuvust, kui enne mannekeeni mudeli projekteerimist kokku sai lepitud. Liikumisulatuse suurendamise järele võib tekkida vajadus ka siis, kui mannekeeni kehaasendite defineerimisel selgub, et asendeid tuleb esialgu planeeritust rohkem või mõne asendi saavutamiseks jääb pisut liikumisulatuses puudu.

Teiseks haldustarkvara ülesandeks on aktuaatorite püsivara uuendamine. See on vajalik nii funktsionaalsuse lisamiseks kui ka töö jooksul ilmnenu tarkvaravigade paikamiseks. Koos püsivaraga uuendatakse ka aktuaatorite konfiguratsiooniväärtused. Käesoleva magistr töö aluseks olevas haldustarkvaras saab aktuaatoreid uuendada ainult ühekaupa.

Viimaseks haldustarkvara suuremaks funktsiooniks on aktuaatorite info vaatamine. Info kuvatakse tabelis, milles on iga aktuaatori kohta kirjas identifikaator, tüüp, seisukord ning UID. Aktuaatori seisukorral on viis võimalikku väärtust:

- *OK* - aktuaator on töökorras,
- *MISSING* - aktuaator ei ole mannekeeniga ühendatud,
- *DUPLICATE* - sama tüübi ja id-ga aktuaatoreid on ühendatud rohkem kui üks,
- *XTRA* - antud id-ga aktuaator ei kuulu mannekeeni aktuaatorite nimekirja,
- *INVALIDTYPE* - füüsilise aktuaatori tüüp ning andmebaasis olev tüübiinfo (antud mannekeeni kohta) on erinevad.

Info on kuvatud kujul nagu on näha tabelist 2.1.

Tabel 2.1: Aktuaatorite info kujutamine haldustarkvaras

ID	Type	Diagnose	UID
11	AC135SA	OK	106430752
12	AC115SA	OK	106430506
...			
93	AC55SA	OK	106430812
94	AC55SA	OK	106430349

2.3 Mannekeeni rakendusliides

Mannekeeni rakendusliides on tarkvara, mis võimaldab mannekeeni juhtida USB-ühendust kasutades. Näiteks võtta sisse erinevaid kehakujusid, uuendada

aktuaatorite püsivara ning lugeda veateateid. Liidest kasutavad kõik tarkvarad, mis juhivad või haldavad mannekeene.

2.4 Veebirakendus *Support Software*

Support Software on veebirakendus, milles hoitakse mannekeenide ja aktuaatoritega seotud infot, näiteks mis tüüpi aktuaatoritest mannekeeni versioon koosneb ning millised füüsilised aktuaatorid on mannekeeni mudelis kasutusel. Antud väärtusi saab veebirakenduses muuta ning neid kasutab PTD.

Support Software'ist on olemas kaks versiooni. Varasem versioon on loodud C#-is ja kasutab .NET teeki ning uuem versioon on implementeeritud PHP-s (Laravel). Vanemat versiooni enam edasi ei arendata aga vajadusel viiakse sisse veaparandusi.

2.5 Andmebaasi API

Andmebaasi API ühendub andmebaasiga ning on liideseks andmebaasi ning teiste rakenduste vahel, mis vajavad oma tööks andmebaasis olevat infot. API-t kasutavad näiteks PTD ja *Support Software*.

2.6 Tarkvara *Test Unit*

Test Unit on käsuarakendus, mis võimaldab mannekeeni kohta näha sama infot, mida kuvab PTD. Lisaks võimaldab see näha täpsemat veainfot ning aktuaatori tarbitavat voolu. Viimase väärtuseks on voolu maksimaalne väärtus liikumise jooksul ning seda infot uuendatakse liikumise lõppedes. *Test Unitiga* saab teha mannekeeni aktuaatoritele koormustesti. Aktuaatoreid liigutatakse eelnevalt määratud koordinaadile ja tagasi senikaua kui test peatakse või ilmneb viga.

3 Nõuded tarkvarale

3.1 Vajadused

Vaja on komplektset tarkvara, mis võimaldab aktuaatorite püsivara ja konfiguratsiooni uuendada, näha veateateid ning testida aktuaatoreid. Viimane on vajalik selleks, et saaks potentsiaalselt probleemsed aktuaatorid välja vahetada regulaarse hoolduse käigus, mis tagab selle, et pildistamises ei teki seisakuid. Näiteks, kui pildistamise käigus aktuaator töötamise lõpetab, siis selle väljavahetamine võib võtta 30 kuni 60 minutit, olenevalt aktuaatori asukohast mannekeenil.

Komplektse tarkvara nõude põhjuseks oli olukord, kus eksisteerib mitu pisemat tarkvararakendust, mis teevad sarnaseid asju, aga kuna üheski ei ole kogu funktsionaalsust, siis oli vaja samaaegselt arendada ja kasutada mitut erinevat programmi. Kuna erinevat tarkvara kasutati sama töö tegemiseks, siis arenduse käigus pidi funktsionaalsuse lisama kõikidesse programmidesse ning vigade või puuduste ilmemisel oli vajalik need ära parandada kõikides tarkvarades.

Käesoleva magistritöö raames loodud tarkvara põhjaks valiti haldustarkvara PTD, kuna selles sisaldus suurim valim reaalselt kasutatavast funktsionaalsusest.

3.2 Funktsionaalsed nõuded

Mannekeeni haldustarkvarale esitatud funktsionaalsed nõuded jagunevad kolmeks: arendajale oluline, tehnilisele toele oluline ning funktsionaalsus, mis on oluline nii arendajale kui ka tehnilisele toele.

3.2.1 Arendajale olulised nõuded

Arendajale oluline funktsionaalsus on järgmine.

Aktuaatorite voolutarbe mõõtmine. On vaja mõõta, kuidas aktuaatorite poolt tarbitav vool muutub, kui mannekeeniga võetakse sisse erinevaid asendeid. Oluline on eristada, kas liikumine toimub suuremast asendist väiksemasse või vastupidi (kas aktuaator liigub sisse või välja). Mõõdetud voolu kuvatakse graafikul, kus x-teljel on aktuaatori koordinaat ning y-teljel voolutarve. Kui vool ületab etteantud piiri, tuleb kasutajat hoiatada. Asendite infot loetakse andmebaasist ning seda saab uuendada kasutades *Support Software*'i.

Aktuaatori konfiguratsiooni muutmine. Aktuaatorite konfiguratsiooniväärtuseid peab saama muuta vastavalt mannekeeni mudelile. Väärtused on defineeritud

andmebaasis ning neid saab muuta läbi *Support Software*'i. Aktuaatorit peab saama mannekeeniga siduda ainult siis, kui selle identifikaatori väärtus on 1021 (tähistab tühja konfiguratsiooniga aktuaatorit).

Erinevate mehaaniliste ja elektrooniliste osade taaskasutamine. Kui aktuaator ei tööta ning seda pole mõistlik parandada või pole antud füüsilise konfiguratsiooniga aktuaatorit enam vaja, peab PTD-s saama valida aktuaatori unikaalse identifikaatori järgi, et see läheb mahakandmisele. Lisaks peab saama nimekirjast valida, millised osad võetakse uuesti kasutusele. Kui ühtegi osa ei ole valitud, kuvatakse vastav hoiatus ning küsitakse kasutajalt kinnitust, et ühtegi osa ei lähe taaskasutusse. Kui UID automaatne lugemine ebaõnnestub, peab saama seda käsitsi lisada. Enne muudatuste tegemist küsitakse kasutaja käest kinnitust, et kas ta soovib varasemalt sellisel mannekeenil ning sellise identifikaatoriga aktuaatorit taaskasutusse võtta.

3.2.2 Tehnilisele toele olulised nõuded

Tehnilisele toele oluline funktsionaalsus on järgmine.

Hoiatuste kuvamine. Kasutajale peab kuvama hoiatusi. Hoiatused peab jagama kolme kategooriasse:

1. hoiatusele peab tähelepanu pöörama järgmise regulaarse hoolduse käigus,
2. mannekeeni saab edasi kasutada, aga seda tuleks teha ainult juhul, kui see hädavajalik on,
3. mannekeeni ei saa edasi kasutada.

Esimesse kategooriasse kuulub aktuaatori läbisõidu piirangu ületamine. Teise kuulub mannekeeni aktuaatorite tüüpide erinevine andmebaasis kirjeldatust ning kolmandasse mannekeeni vearežiimis olemine või aktuaatori enesetesti mitteläbimine. Tarkvara peab kasutajat selgelt teavitama, kui kriitilise probleemiga on tegu.

3.2.3 Ühine osa

Funktsionaalsus, mida vajavad nii arendajad kui ka tehniline tugi, on järgnev.

Mannekeeni valimine. Alati ei ole arvutiga ühendatud ainult üks mannekeen. Kasutaja peab saama valida, millist mannekeeni ta haldab ning programmi käivitamisel ei tohi ükski mannekeen valitud olla.

Aktuaatorite info vaatamine. Lisaks tabelis 2.1 kirjeldatud infole tuleb kuvada UID kuueteistkümnendsüsteemis, kas ja millises veaolukorras on aktuaator ning püsivara versioon.

Mannekeenile väärtuste kirjutamise piiramine. Enne iga tegevust, mis uuendab aktuaatori konfiguratsiooniväärtuseid mannekeenil, peab kontrollima, kas lukuvõti on keeratud suletud asendisse. Kui lukk on avatud asendis ei tohi saada väärtuseid kirjutada.

Aktuaatori püsivara uuendamine. Peab saama uuendada kõikide aktuaatorite püsivara, kasutamata täiendavat programmeerimisriistvara (näiteks JTAGICE3 [3]). Peab saama valida, kas uuendatakse ainult ühte aktuaatorit või kõiki korraga. Vaikimisi peab kasutama andmebaasis olevat kõige uuemat püsivara, kuid soovi korral on võimalik püsivara uuendada, kasutades arvutis olevat hex-faili. Kasutajale teavitatakse alati, kui saadaval on uuem püsivara versioon ning aktuaatoreid peaks uuendama.

Aktuaatori konfiguratsiooniväärtuste uuendamine. Võimalus uuendada aktuaatorite konfiguratsiooniväärtuseid. Väärtused asuvad *Support Software*'i andmebaasis. Aktuaatoreid peab saama uuendada nii ühekaupa kui ka kõiki korraga.

***Support Software*'i andmebaasiga ühendumine.** Kogu programmi jaoks vajalik info paikneb andmebaasis. Infot on vaja lugeda ning muuta. Andmebaasis hoitakse näiteks aktuaatorite käigupiirangu väärtuseid ning millist tüüpi aktuaatorid on mannekeenil kasutuses.

Mannekeeni füüsilise konfiguratsiooni lugemine. Võimalus lugeda infot, millist tüüpi ning milliste konfiguratsiooniväärtustega aktuaatorid mannekeeniga ühendatud on.

Kasutatava andmebaasi valimine. Programmi käivitamisel saab kasutaja valida, kas kasutatakse testimise andmebaasi või töös olevat andmebaasi.

Kasutajad ning kasutajarollid. Tarkvara kasutamiseks on vaja sisestada kasutajanimi ning parool. Vastavalt rollile on kasutajale lubatud kindlad tegevused. Kasutajad ning rollid on kirjeldatud ja muudetavad *Support Software*'is. Autentimise ja rollide eesmärgiks on vältida mannekeeni füüsilist kahjustumist väärkasutamisel. Haldustarkvaraga on võimalik sooritada tegevusi, mis võivad potentsiaalselt mannekeeni lõhkuda, kui funktsionaalsust oskamatult või pahatahtlikult kasutada. Lisaks antud tarkvarale on vastavad kontod kasutuses ka teistes rakendustes, mida kasutatakse mannekeeni osade testimiseks.

Tegevuste ajalugu. Kõikide tegevuste kohta, mis muudavad andmebaasi sisu või mannekeeni tarkvara, säilitatakse info. Iga tegevuse kirjes sisaldub kasutajanimi, aeg ning kommentaar, mis põhjendab antud toimingut.

Andmebaasi logi kirjutamise ajutine peatamine. Tegevuste ja veateadete logimist andmebaasi saab vajadusel välja lülitada, et vältida testimisel andmebaasi liigse info kirjutamist.

Tarkvarauuendused. Tarkvara annab kasutajale märku, kui saadaval on uus versioon ning palub kasutajal see alla laadida.

3.3 Mittefunktsionaalsed nõuded

Järgnevalt on kirjeldatud tarkvarale esitatud mittefunktsionaalsed nõuded.

- Tarkvara nimeks on PTD (*Programming Tool for Developers*).
- Tarkvara peab töötama Windowsi operatsioonisüsteemil alates versioonist 7.
- Kasutajaliides on inglisekeelne.

- Välja on vaja töötada testimisplaan, mis annab teada, kas kogu nõutud funktsionaalsus töötab. Testid on kirjeldatud sellise täpsusega, et teised meeskonnaliikmed oskavad tarkvara testida.
- Soovitud funktsionaalsuse kasutamiseks ei tohi kuluda üle viie hiirekliki .
- Kasutajaliides peab olema intuitiivne, tarkvara kasutamiseks ei ole vaja koolitust läbida.
- Käitades ei tarbi tarkvara muutmälu rohkem kui 50 MB.
- Ühildub *Support Software*'iga.
- Töötab tagasiulatuvalt erinevate mannekeeni versioonidega alates V7.1-st.
- Kogu lähtekood on dokumenteeritud.
- Jälgida tuleb valitud programmeerimiskeele standardeid ning koodimistavasid.
- Andmevahetus haldustarkvara ning andmebaasi vahel peab olema krüpteeritud.
- Iga mannekeeni versioonil on erinevad aktuaatorite konfiguratsiooniväärtused. Ühe mannekeeni väärtuseid ei tohi olla võimalik kirjutada teise mannekeeni aktuaatoritele.

3.4 Olemasoleva haldustarkvara probleemid

Olemasolev tarkvara PTD jääb uue haldustarkvara arendamise põhjaks. Enne uue funktsionaalsuse lisamist on vaja parandada mittetöötav funktsionaalsus. Järgnevalt on välja toodud arendusel aluseks võetud haldustarkvara probleemsed kohad.

- Aktuaatorite infot sisaldava tabeli kerimine on aeglane ning hüplik, lisaks ei saa infot tabelis sorteerida ega sellest väärtuseid kopeerida. Viimane on oluline, sest vahepeal on vaja aktuaatoreid *Support Software*'ist UID järgi otsida.
- Aktuaatorite püsivara ja konfiguratsiooniväärtuste uuendamist saab teha ainult ühekaupa. Kõikide mannekeeni küljes olevate aktuaatorite uuendamise võimekus on olemas olnud, aga hetkel see ei tööta.
- Aktuaatorite konfiguratsiooni uuendamisel uuendatakse ka püsivara. Need on vaja üksteisest eraldada, sest konfiguratsiooni uuendatakse tihedamini kui püsivara. Samuti esineb olukordi, kus arenduses või testimise eesmärgil on vaja uuendada konfiguratsiooni, kuid kasutada kõige viimase püsivara versiooni asemel mõnda vanemat versiooni.
- Püsivara uuendades kirjutatakse kõigepealt uus tarkvara aktuaatorisse ning pärast seda üritatakse andmebaasis väärtuseid uuendada. Kui mingil põhjusel andmebaasiga ühendust ei saada või väärtuste uuendamine ebaõnnestub, siis ei ole andmebaasis olev info enam ajakohane. See on probleemiks, kuna andmebaasis olevat infot kasutatakse tuvastamiseks, millised mannekeenid vajavad uuendamist pärast uue püsivara versiooni valmimist.

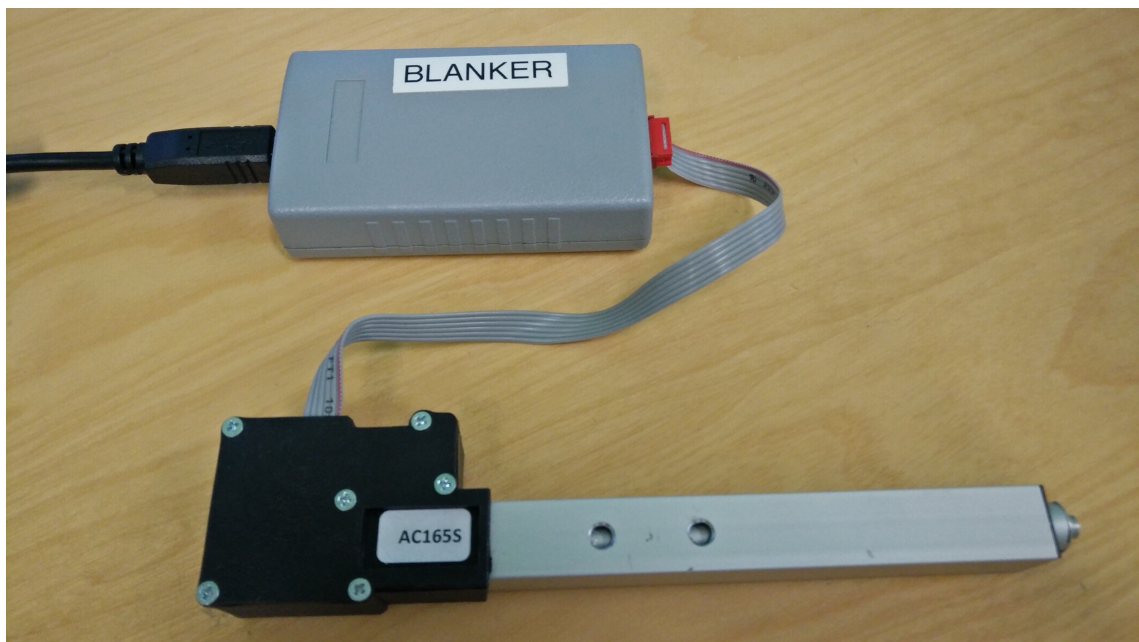
3.5 Kasutusjuhud

Järgnevalt on toodud haldustarkvara peamised kasutusjuhud, mis on olulised nii mannekeeni arendamisel kui ka pildistamisel.

3.5.1 Aktuaatori konfiguratsiooniväärtuste kustutamine (puhastamine)

Aktuaatori konfiguratsiooni on vaja kustutada juhul, kui aktuaator on varasemalt olnud mõne mannekeeni küljes kasutuses ning seda soovitakse taaskasutada. Taaskasutuse korral paigaldatakse aktuaator mõnele teisele mannekeenile või pannakse see varuosade lattu. Konfiguratsiooni saab kustutada ainult tööriistaga *blanker*.

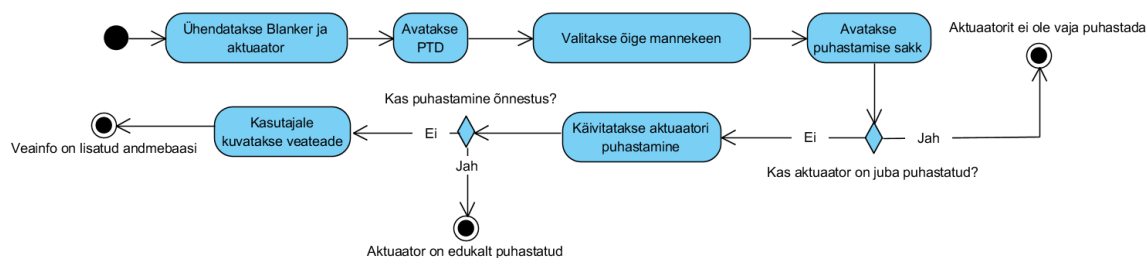
Töö teostamiseks on vajalikud *blanker*, kasutuses olnud aktuaator ning arvuti, kuhu on paigaldatud mannekeeni haldustarkvara. Kõigepealt ühendatakse osad omavahel: *blanker* arvuti külge ning aktuaator *blankeri* külge. Ühendamine on kujutatud joonisel 3.1. Juba puhastatud aktuaatorit teistkordselt puhastada ei saa. Kõigepealt muudetakse ära aktuaatori identifikaator ning pärast seda uuendatakse andmebaasi infot: uuendatakse identifikaatorit ning aktuaatori kirjelt eemaldatakse seos mannekeeniga. Kui ühetegi viga ei esine, antakse kasutajale teada, et konfiguratsiooni kustutamine õnnestus. Vastasel juhul näeb veateadet töö ebaõnnestumise infoga. Protsessi voodiagramm on toodud joonisel 3.2.



Joonis 3.1: Aktuaatori ühendamine *blankeriga*.

3.5.2 Aktuaatori konfiguratsiooniväärtuste uuendamine

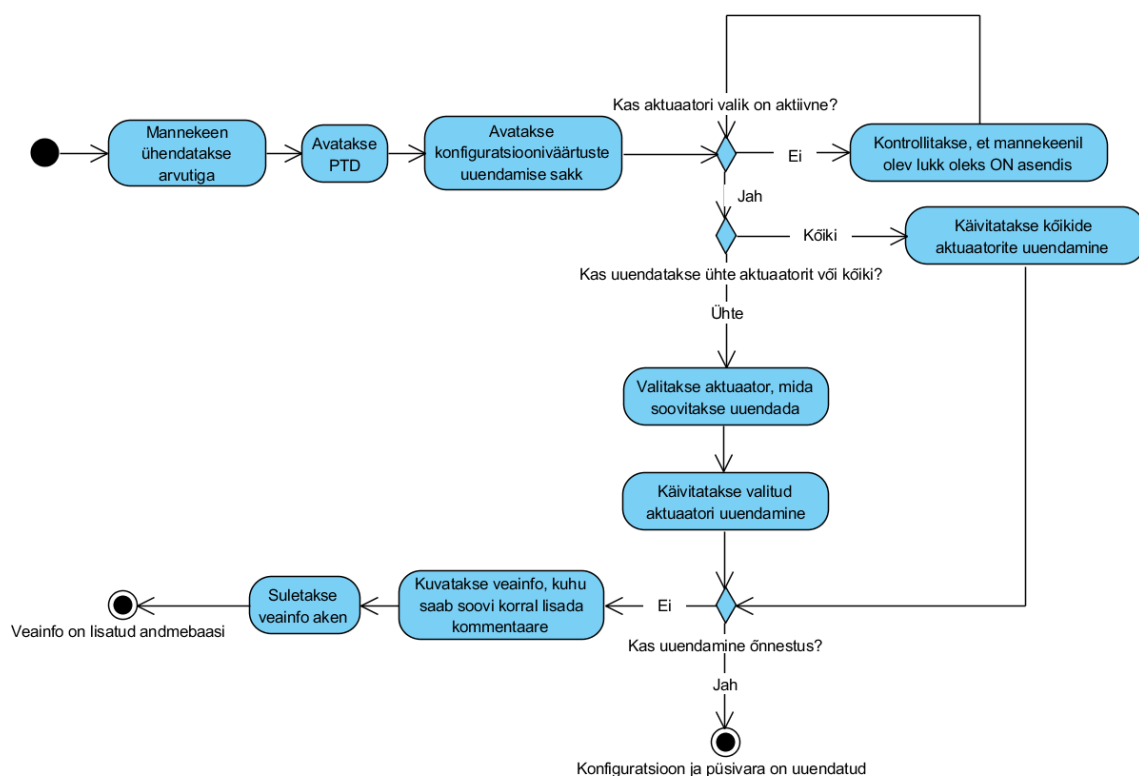
Andmebaasis hoitakse infot aktuaatori konfiguratsiooni kohta. Konfiguratsiooniväärtuseid on võimalik muuta läbi *Support Software*'i. Aktuaatori konfiguratsiooniväärtuseid uuendatakse juhul, kui *Support Software*'is on vastavaid



Joonis 3.2: Aktuaatori konfiguratsiooniväärtuste kustutamine.

väärtuseid uuendatud, mida tehakse enamasti mannekeenide arendamise etapis.

Kõigepealt ühendatakse mannekeen arvutiga ning seejärel avatakse PTD. Pärast õige mannekeeni valimist rippmenüüst avatakse konfiguratsiooniväärtuste uuendamise sakk. Üksiku aktuaatori uuendamisel valitakse kõigepealt aktuaator ning seejärel käivitatakse uuendamise protsess. Kõikide aktuaatorite uuendamiseks on eraldi valik *Update all*. Uuendamise kulgu (kui palju aktuaatoreid uuendatud on) näeb allosas olevalt progressiribalt. Vea tekkimisel teavitatakse kasutajat ning kirjutatakse veainfo andmebaasi. Kogu tegevust kirjeldab joonisel 3.3 olev voodiagramm.



Joonis 3.3: Aktuaatori konfiguratsiooniväärtuste uuendamine

3.5.3 Aktuaatori püsivara uuendamine

Püsivarasse lisatakse aeg-ajalt uut funktsionaalsust ning parandatakse vigu, mis testimise ja kasutamise käigus välja tulevad. Seetõttu on vaja aktuaatorite püsivara uuendada.

Kõigepealt ühendatakse mannekeen arvutiga ning avatakse PTD. Pärast sisselogimist valitakse rippmenüüst mannekeeni versioon ning seejärel avatakse sakk *Firmware Update*. Kui on tarvis uuendada ainult ühe aktuaatori püsivara, saab rippmenüüst selle valida ning käivitada uuendamisprotsessi. Kõikide aktuaatorite uuendamiseks on eraldi valik *Update all*. Viimasel juhul on võimalik progressiribalt näha, millise aktuaatori juures järg on. Protsess on väga sarnale konfiguratsiooni uuendamisele, mida kirjeldab joonis 3.3.

3.6 Kasutajarollid ja õigused

Mannekeeni elutsükli jooksul puutuvad mannekeeniga kokku erinevate tööülesannetega kasutajad. Haldustarkvara on abivahend mitmete mannekeenidega seotud tööülesannete täitmisel. Haldustarkvara arenduse tarvis on defineeritud kasutajarollid ning iga rolli jaoks vajalikud tegevused. Erinevad võimalikud funktsioonid, mida tarkvaras kasutada saab, on järgmised.

Info vaatamine - kõikide aktuaatorite info vaatamine. Väärtused on kirjeldatud peatükis 3.2.3.

Konfiguratsiooni kustutamine - aktuaatori ettevalmistamine uuel mannekeenil kasutamiseks.

Konfiguratsiooni kustutamine *blankerita* - aktuaatori ettevalmistamine uuel mannekeenil kasutamiseks ilma kindla tööriistata (*blanker*).

Konfiguratsiooni uuendamine kõigile - kõikide valitud mannekeeniga ühendatud aktuaatorite konfiguratsiooniväärtuste uuendamine.

Konfiguratsiooni uuendamine valitutele - valitud aktuaatorite konfiguratsiooni-väärtuste uuendamine.

Püsivara uuendamine - kõikide valitud mannekeeniga ühendatud aktuaatorite püsivara uuendamine viimasele versioonile.

Püsivara vahetamine - aktuaatori püsivara vahetamine valitud versioonile.

Konfiguratsiooni hoiatuse vaatamine - hoiatuse kuvamine, kui andmebaasis olev info ei kattu mannekeeni füüsilise konfiguratsiooniga.

Aktuaator taaskasutusse - võimalus võtta kindla aktuaatori komponente taaskasutusse.

Kasutajarollide vahetamine - võimalus vahetada kasutajarolle, et rollide piiranguid saaks mugavalt testida.

Aktuaatori mahakandmine - Komplektse aktuaatori info kustutamine andmebaasist ning valitud osade taaskasutamine.

Aktuaatori läbisõidu hoiatuse vaatamine - hoiatuse kuvamine, kui aktuaatori läbisõit ületab etteantud piiri.

Rakenduses on viis erinevat rolli: tarkvara, elektroonika ja mehaanika arendaja ning tehniline tugi ja noorem tehniline tugi. Tarkvara arendaja ülesanneteks on silla ja aktuaatori püsivara arendamine, testimine ja parandamine. Elektroonika arendaja disainib, testib ja parandab mannekeenide juhtelektroonikat. Mehaanika arendaja disainib, testib ja parandab mannekeeni mehaanikat, lisaks koostab mannekeene. Tehniline tugi hooldab ja parandab mannekeene. Noorem tehniline tugi kontrollib kasutuses olevaid mannekeene ning annab leitud vigadest tehnilisele toele teada. Õiguste jaotumine on toodud lisas 1, tabelis 8.1.

4 Eeltöö ja uuringud

Järgnevalt on välja toodud, mida on vaja täpsustada, enne kui saab tarkvara arendamisega alustada.

4.1 Intuiitiivne kasutajaliides

Üks nõuetest oli, et kasutajaliides peab olema intuiitiivne. Et seda täita, on kõigepealt vaja määrata intuiitiivse kasutajaliidese tingimused.

Kasutajaliides on intuiitiivne, kui kasutajad mõistavad liidese elementide käitumist ja mõju ilma, et nad peaks katsetama, abi paluma või väljaõpet vajama. Liidese kasutamisel tugineb inimene oma varasematel kogemustel ning neid kasutades oskab hinnata, kuidas kindel liidese osa toimima peaks. Näiteks, kui miski näeb välja nagu nupp, siis kasutaja eeldab, et seda saab vajutada ning pärast seda juhtub midagi. Samuti nähes linki (internetilehel), teab kasutaja, et sellele vajutades saab uuele lehele minna. [11]

Intuiitiivset kasutajaliidest on keeruline luua, sest kasutajad on erinevate varasemate kogemustega. See, mis on ühe kasutaja jaoks intuiitiivne ja iseenesest mõistetav ei pruugi seda olla teise jaoks. [14]. Näiteks Windowsi kasutajale on iseenesest mõistetav, et töölaual olevaid programme saab käivitada hiire topelt vasakkõpsuga. Samas kui kasutaja on harjunud kasutama Linuxi KDE Plasma töölauda [20], kus osades versioonides kasutatakse programmide avamiseks vaikimisi vasakklõpsu, siis topeltklõps ei ole tema jaoks intuiitiivne.

E. McKay defineerib intuiitiivse kasutajaliidese järgmiselt: kasutajaliides on intuiitiivne, kui see vastab vähemalt osaliselt järgmistele nõuetele: [11]

- Kasutajaliides kasutab visuaalseid vihjeid, mis aitavad mõista, mis parajasti toimub. Liides on sarnane füüsilistele kasutajaliidestele, mille kasutamise oskus on juba inimesel olemas.
- Funktsionaalsus on etteaimatav. Liidese elemendi, näiteks nupu, kasutamisel on tulemus oodatav ning etteaimatav. Kasutaja ei pea katsetama ega arutlema, et tulemust mõista.
- Soovitud tulemuse saavutamiseks peab kulutama võimalikult vähe energiat. Kui kasutaja soov on selge, siis liidest kasutades saab soovitud tulemused esimese korraga. Vältima peab olukordi, kus kasutaja peab läbi proovima erinevaid võimalusi, et soovitud toiminguid sooritada.

- Toimingut tehes annab liides kohese tagasiside, et käsku täidetakse ning kas tulemus oli edukas või mitte.
- Kasutajale antakse tehtud vead andeks. Kogemata valet toimingut tehes on võimalik seda mugavalt peatada või tagasi võtta.
- Kasutajaliideses saab ringi liikuda, ilma et peaks kartma äraeksimist või ootamatut käitumist.
- Kasutajaliidese kasutamine ei tekita inimestes negatiivseid tundeid ning liidese kasutamine tekitab nendes rahulolu.

Haldustarkvara arendamisel on vaja algne kasutajaliides üle vaadata ning hinnata, milliseid osasid on vaja muuta, et liidest intuiitiivsemaks muuta. Lisaks on vaja otsustada, millised kasutajaliidese intuiitiivsuse nõuded on haldustarkvara juures olulised.

4.2 Tarkvara testimine

Tarkvara testimiseks on vaja uurida millised meetodid on olemas ning pärast seda valida välja sobiv(ad) antud tarkvara jaoks. Tarkvara testimiseks on neli suuremat meetodit: musta kasti, valge kasti, halli kasti ning juhuslik testimine [18].

Musta kasti testimise puhul ei tea testija, kuidas funktsionaalsus lähtekoodis teostatud on. Võimalik on teha nii funktsionaalseid kui ka mitte funktsionaalseid teste, kuid üldjuhul tehakse esimesi. Testimisega üritatakse leida järgmiseid vigu: valesti töötavad või puuduvad funktsioonid, liidese vead, andmestruktuuri ja andmebaasiühenduse vead, käitumise või jõudluse probleemid ning objektide loomise ja ressursi vabastamise vead. Selle meetodi eelisteks on esiteks testide tegemine kasutaja vaatepunktist, mis võimaldab leida kõrvalekaldeid nõuetest. Teiseks ei pea testija ennast kursi viima ei arenduskeelega ega tarkvara lähtekoodiga. Lisaks saab teste teha arendajatest eraldi, mis suurendab testide objektiivsust. Viimaseks saab testide loomisega pihta hakata kohe, kui nõuded on kinnitatud. Puudusteks on, et testida saab ainult väga väikese osa kõikidest võimalikest sisenditest ning seega võivad osad probleemid avastamata jääda. Testide kirjutamiseks on vaja põhjalikke nõuded, aga paljude projektide puhul on nõuded puudulikud. [16]

Valge kasti testimise puhul, vastupidiselt musta kasti testimisega, on testijale teada tarkvara sisemine struktuur. Näiteks veebirakenduse puhul kindla sisendivälja testimiseks on vaja kõigepealt määrata kõik võimalikud lubatud ja keelatud sisendid ning pärast seda on vaja kontrollida, kas väljund vastab ootustele. Sisendite ja oodatud väljundite väärtused leitakse lähtekoodi uurides. Selle meetodi eelisteks on varajane testimine, funktsioone saab testida kohe, kui need on valmis ning ei pea ootama kasutajaliidese valmimise järel. Võrreldes musta kastiga on testimine põhjalikum, mis annab suurema tõenäosuse, et leitakse rohkem vigu. Puudusteks on, et testijatelt on nõutud rohkem eelteadmisi ning nad peavad tuttavad olema nii arenduseks kasutatava programmeerimiskeele iseärasustega kui ka lähtekoodi endaga. Kui funktsioonide implementatsioon muutub tihti, siis testide ajakohasena hoidmine nõuab palju ressursi. Lisaks on testid tihedalt seotud konkreetse rakendusega, seega on vähe valmis tööriistu, mida saaks kasutada, et testide loomist kiiremaks muuta. [19]

Halli kasti testimises kombineeritakse kaks eelnevalt kirjeldatud testimise meetodit.

Testija teab koodi ülesehitust ning kasutab seda teadmist testide kirjutamisel, kuid testitakse mooduli avalikke meetodeid ja funktsioone. Põhiliseks eeliseks on see, et luua saab paremaid funktsionaalteste võrreldes musta kasti testimisega. [17]

Juhusliku testimine puhul ei looda testimise plaani ega kasutada dokumentatsiooni. Teste tehakse juhuslikult ning eesmärgiks ei ole saada konkreetseid tulemusi. Sellise testimise puhul on vigu keeruline taastekitada, aga samas võib avastada vigu, mis ühegi teise testimisega välja ei tuleks, sest seal järgitakse ranget plaani. Mida leidlikum ja püsivam testija on, seda paremaid tulemusi juhuslik testimine annab. [15]

PTD puhul on oluline, et funktsionaalsus töötaks ning seda oleks mugav kasutada. Funktsionaalsuse testimiseks saab kasutada nii musta kui ka valge kasti meetodit. Mannekeeni arendusmeeskond on väike ning tarkvaraarendajatel on koormus kõige suurem. Töökoormuse hajutamiseks tegelevad nad ainult arendamise ja esialgse testimisega ning arendatud tarkvara põhjalikuma testimisega tegelevat teised meeskonnaliikmed. Nende kogemused arenduskeele ja tarkvaraarendamisega on piiratud, seega on kõige mõistlikum kasutada musta kasti meetodit.

4.3 Vigade ennetamine ja ennustamine

Tagamaks, et mannekeenid töötavad pildistamisel stabiilselt, tehakse igale robotile regulaarselt hooldust. Kui mannekeen on aktiivselt kasutuses, siis kontrollitakse robotit iga 30 päeva tagant. Kui mannekeeni töös esineb tõrkeid, siis teostatakse ebaregulaarset hooldust. Hoolduse käigus kontrollitakse, kas juhtprogramm saab mannekeeniga ühendust ning viiakse läbi enesetest. Seejärel võetakse sisse tellijaga eelnevalt kokku lepitud asendid, mida kasutatakse mannekeeni asendite korratavuse testimiseks. Iga mannekeeni kehaasendi puhul võetakse kokkulepitud kohtadest mõõdud ning võrreldakse neid etaloniga. Mõõtude erinevus etalonist annab märku võimalikust probleemist. Seejärel kontrollitakse mehaanikat ja aktuaatoreid: kas kõik detailid ja kaablid on korrektselt kinnitatud. Pärast seda kontrollitakse roboti välist ilmet: kas katted on korrektselt paigaldatud ja kinnitatud ning ega kusagil kortse ei ole. Viimaks võetakse sisse kõik pildistamiseks kasutatavad asendid, et veenduda nende korrektsuses.

Hetkel kasutuses oleva testimisprotseduuri üheks suuremaks puuduseks on kõigi mannekeenide testimine sama intervalliga, olenemata reaalsest kasutskoormusest või vigade esinemise sagedusest. On vaja uurida, milliseid meetodeid sarnastes rakendustes kasutatakse ning kas neid saab rakendada mannekeeni juures, et ennetada tekkivaid vigu.

4.3.1 Ülevaade

Vigu ennetavat hooldust tehakse kindlate ajavahemike järel. Hoolduste intervallid määratakse varasemast kogemusest: kui tihti varasemalt on probleeme esinenud ning mis on antud seadme eeldatav tööaeg enne korrapärast hooldust. [6]

Vigu ennustav hooldus võimaldab hinnata, millisel hetkel peaks robotit hooldama või varuosasid vahetama, et tagada robotite stabiilne töötamine. Vigade ennustamiseks kogutakse andmeid tavaliselt roboti töö käigus, et võimalikult vähe häirida tegevusi, mida robot sooritama peab. Robotite seisukorra hindamiseks kasutatakse näiteks infrapuna- ja

vibratsiooniandureid ning visuaalset kontrolli. Lisaks kontrollitakse ka teisi parameetreid, mis võivad antud rakenduses olulised olla. [2, 10]

Infrapunaandureid kasutatakse põhiliselt elektriliste probleemide avastamiseks. Andurid annavad lihtsasti ja kiiresti tulemuse ning need ei pea mõõtmistulemuse saamiseks seadmega kontaktis olema. Vibratsioonianduritega mõõdetakse roboti poolt tekitatava vibratsiooni sagedust ning amplituudi. Kui hõõrdumine suureneb või mõnele osale rakendub liiga suur jõud, siis muutub ka vibratsiooni sagedus ja amplituud. Mõõdetud vibratsiooni põhjal saab hinnata, kas ning milline varuosa väljavahetamist vajab. Visuaalse kontrolli puhul hinnatakse robotite välist ilmet ning otsitakse näiteks paindumisi, murdumisi, pragusid ja korrosiooni. Eesmärgiks on veenduda, ega ükski osa ei näe tavapärasest erinev välja. [2, 10]

4.3.2 Vigade ennustamine mannekeenil

Mannekeenidel kasutatakse hetkel vigu ennetavat hooldust. Samas ei kasutata hooldusintervallide määramiseks aktuaatori eeldatavat tööaega. Tööaega on keeruline määrata, sest aktuaatoritel on erinevad käigupikkused ja -piirangud ning olenevalt sellest, milliseid kehakujusid pildistamiseks kasutatakse, on aktuaatorite maksimaalne liikumine erinev. Näiteks mõni aktuaator liigub täisulatuses sisse ja välja, samas teine liigub ainult mõne sentimeetri. Lisaks on aktuaatorite töökoormus erinev. Õlavöötmes on sama tüüpi aktuaatori koormus suurem võrreldes keha alaosaga, sest aktuaator peab mannekeeni kuju muutmiseks rakendama suuremat jõudu, kuna mannekeeni kate avaldab rohkem jõudu ning kate pindalaühiku kohta on vähem aktuaatoreid.

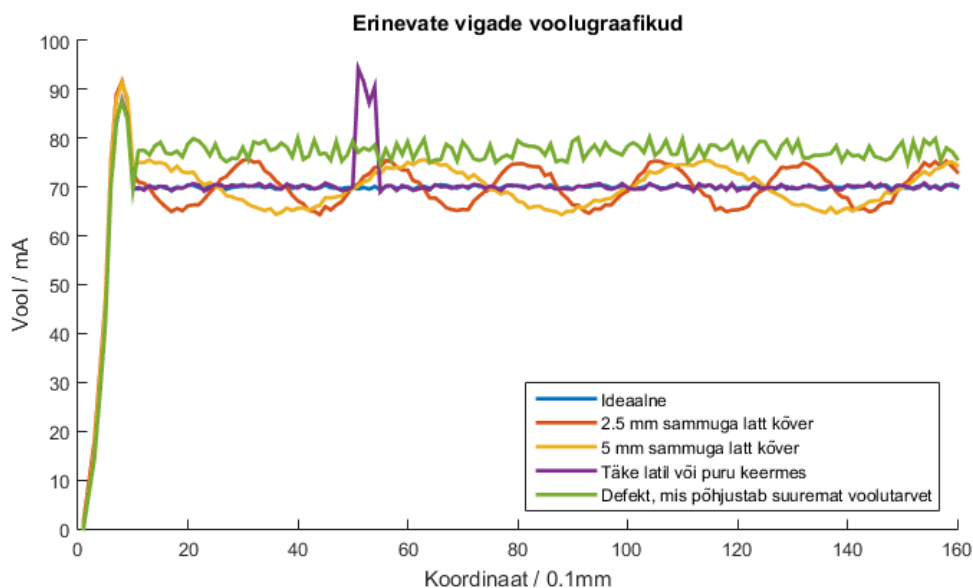
Hooldusvälpade määramiseks võiks kasutada varasemalt esinenud vigade hulka. Näiteks vaadata, kui palju on eelneva 10 päeva jooksul vigu esinenud ning vastavalt sellele määrata järgmise korralise hoolduse aeg. Samas pole see lähenemine ideaalne, kuna mannekeene ei kasutada pidevalt. Ühes kuus võidakse kasutada ainult meesmannekeene ning teises kuus naismannekeene. Antud juhul ei ole korrektne määrata meesmannekeenide vigade põhjal naismannekeenide hooldusgraafikut, sest esiteks ei ole neid kuu aega kasutatud ning teiseks on naismannekeen ehituse poolest keerukam.

Aktuaatorite poolt tarbitav vool ühe käigutsükli vältel on parameeter, mis näitab aktuaatori koormust. Näiteks kui kahe sama tüüpi aktuaatori tarbitavad voolud on sarnased, siis on sarnane ka nende koormus. Kui aktuaatorite voolutarbimine päevade lõikes kasvab, siis annab see märku probleemist. Voolutarvete mõõtmine sobib hooldusvälpade määramiseks ka ebaregulaarse kasutamise korral. Piisab sellest, kui enne mannekeeni kasutamist mõõta kõikide aktuaatorite voolud maksimaalse liikumise ulatuses ning see annab ülevaate, millises seisus aktuaatorid ja mannekeen on.

Aktuaatori voolutarvet mõõtes saab hinnata selle seisukorda ning vajadusel aktuaator ennetavalt välja vahetada. Aktuaatorite voolutarbe infot saadetakse kogu liikumise vältel ning iga liigutud 0.1 mm kohta saadetakse hetkel tarbitava voolu hulk. Voolutarbe graafikuid jälgides (joonis 4.1) võiks tuvastada näiteks järgnevaid vigu: keermelatt on kõver, latil on mingi osa kannatada saanud või keermelatt vahel on metallipuru, juhik on liiga suur või otsajuhik on liimist lahti tulnud, määret on liiga palju või vähe, ning aktuaatori koormus on tavapärasest suurem. Aktuaatori osasid testitakse enne selle koostamist, aga kõik defektid ei pruugi testimise käigus välja tulla, seetõttu on vajalik

üksikkomponentide vigu ka kasutamise käigus tuvastada. Samuti ei aita aktuaatorite üksikute osade testimine aktuaatori koostamise käigus tekkivate või kasutamisel tekkinud kulumisvigade tuvastamisel.

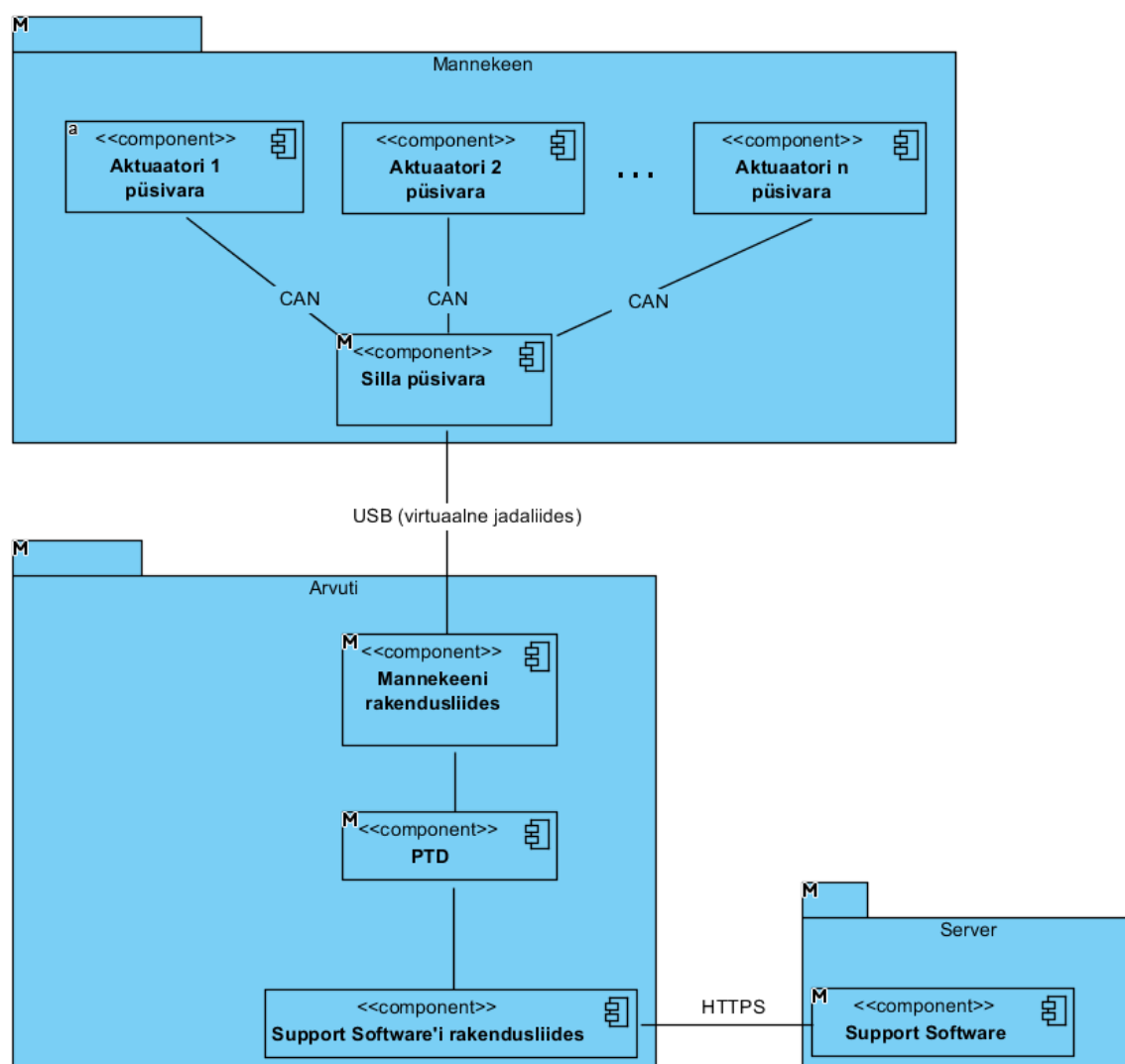
Joonisel 4.1 on kujutatud aktuaatori eeldatavad voolutarbed nii ideaalsel juhul kui ka eri tüüpi vigade korral. Aktuaatorites kasutatakse kahe sammuga keermelatte: 2,5 mm ja 5 mm. Kui need on kõverad, siis voolugraafikul näeme kas iga 2,5 mm või 5 mm tagant kõrgemaid voolu väärtusi. Perioodiliselt kõrgemat voolutarvet võivad põhjustada ka viltu liimitud hammasrattad või hammasratta kahjustused. Kui keermelatil on kahjustus või on keeme vahele puru sattunud, siis paistab see välja ühe või mitme kohana, kus on suurem voolutarbimine. Keskmisest suuremat voolutarvet võib põhjustada määrde liiasus või vähesus, liiga tihe juhik, lahtine otsajuhik või on aktuaatori koostamise käigus midagi valesti või viltu kokku pandud. Suurema voolutarbe põhjuseks võib olla ka kulumine.



Joonis 4.1: Aktuaatori voolugraafikute eeldatav välimus vigade korral. Graafikul on kujutatud ideaalset olukorda, kõverat ja kahjustatud keermelatti või puru keermes ning üldist viga, mis põhjustab suurenenud voolutarvet.

5 Tarkvaraarendus

Haldustarkvara arendus sõltus teistest tarkvarakomponentidest, mida oli vaja haldustarkvaraga koos töötamiseks täiendada. Järgnevalt on lisaks haldustarkvara PTD arendustöödele välja toodud ka teised komponendid, milleta ei saa PTD-d kasutada. Peatükis 2 on kirjeldatud, milline oli tarkvara seis enne magistritöö alustamist ning käesolev peatükk kirjeldab, mida muudeti ja lisati. Joonisel 5.1 on näha komponentide omavahelised seosed.



Joonis 5.1: Tarkvarakomponentide omavahelised seosed

Arendustöö oli jagatud osadesse, mida testiti järk-järgult mannekeeni arendusmeeskonna liikmete poolt, kes on haldustarkvara PTD lõppkasutajad. Selline lähenemine andis võimaluse lisaks tarkvara vigade otsimisele ja parandamisele koguda ka lõppkasutaja tagasisidet ning seda arvesse võtta haldustarkvara arendustöö käigus.

5.1 Mannekeenide haldustarkvara

Järgnevalt on toodud uues mannekeeni haldustarkvaras PTD programmeeritud funktsionaalsuse kirjeldus. Programmi kasutajaliideses on tegevused grupeeritud sakkide alla nimedega *Diagnostics*, *Testing*, *Blanking*, *Node Installation*, *Configuration Update*, *Firmware Update* ning *Recycle* (nt joonis 5.2). Lisaks on välja toodud ka programmi kasutajaliides funktsionaalsuse arendustöö kirjeldus.

5.1.1 Kasutaja sisselogimine

Töö aluseks olevas PTD versioonis oli kasutamise jaoks vaja ainult tarkvara ennast ning internetiühendust. See andis igale tarkvara omavale isikule võimaluse mannekeenide haldamiseks. Samuti ei logitud mannekeenide juures tehtud haldustöid andmebaasi, mistõttu puudus terviklik ülevaade mannekeeni juures tehtust. Uude PTD versiooni on lisatud kasutaja autoriseerimine, kus sisselogimiseks kasutatakse *Support Software*'i kasutajate infot. Iga kasutaja on seotud ühe rolliga ning igal rollil on kindlad õigused (vt lisa 1, tabel 8.1). Haldustarkvarasse sisse logitud kasutajale kuvatakse talle määratud roll programmi alumises paremas nurgas. Tarkvara arendaja saab erinevate rollide vahel valida, tehes hiirega vasakklõpsu rolli nime peal. See avab uue akna, milles saab kasutajarolli valida.

5.1.2 Sakk *Diagnostics*

Joonisel 5.2 on kujutatud sakk *Diagnostics*. Siia on lisatud veateadete, püsivara versiooni ning kuueteistkümnendsüsteemis UID veerg. Lisaks on muudetud tabeli loomist – *ListView* [13] on asendatud *DataGridView*'ga [12]. Muutatus oli vajalik kasutajamugavuse tõstmiseks, sest haldustarkvara algversioonis olid järgmised probleemid:

- ridasid ei saanud valida ega esile tõsta,
- tabelis olevaid kirjeid ei saanud kopeerida,
- tabeli kerimine oli aeglane ja hüplik,
- kasutaja ei saanud tabeli kirjeid veergude kaupa sorteerida.

Püsivara versioon ning kuueteistkümnendsüsteemis UID muudavad aktuaatorite ja silla püsivara ning mannekeeni rakendusliidese testimist ja silumist lihtsamaks, sest nii EEPROM-is kui ka CAN-siinil on andmed esitatud baitide kaupa kuueteistkümnendsüsteemis. Lisaks annab versiooniinfo kiire ülevaate, kas aktuaatorite püsivara vajab uuendamist.

Veerus *Diagnose* kuvatakse mannekeeni veainfot. Lisaks peatükis 2.2 kirjeldatud staatusele kuvatakse selles tulbas uut veateadet *SS_MISSING*, mis tähendab, et aktuaatori

PTD v2.0.0-test.5

Found Mannequins

V25.1

Diagnostics

Testing

Blanking

Node Installation

Configuration Update

Firmware Update

Recycle

Id	Type	Diagnose	Uid	Firmware	Error	UID (hexadecimal)
11	AC135SA	OK	106430752	2.3	Safe mode	0x06580120
12	AC115SA	OK	106430506	2.5	Safe mode	0x0658002A
13	AC145SA	OK	105119761	2.5	Safe mode	0x06440011
14	AC145SA	OK	105119758	2.5	Safe mode	0x0644000E
15	AC115SA	OK	106430568	2.5	Safe mode	0x06580068
16		MISSING	0			0x00000000
21	AC145SA	OK	105119757	2.5	Safe mode	0x0644000D
22	AC115SA	OK	106430477	2.5	Safe mode	0x0658000D
23	AC145SA	OK	105119756	2.5	Safe mode	0x0644000C
24	AC145SA	OK	105775162	2.5	Safe mode	0x064E003A
25	AC115SA	OK	106430527	2.5	Safe mode	0x0658003F
26	AC145SA	OK	105119753	2.5	Safe mode	0x06440009

Safe Mode NOT Overiden

SAFE MODE

Software developer

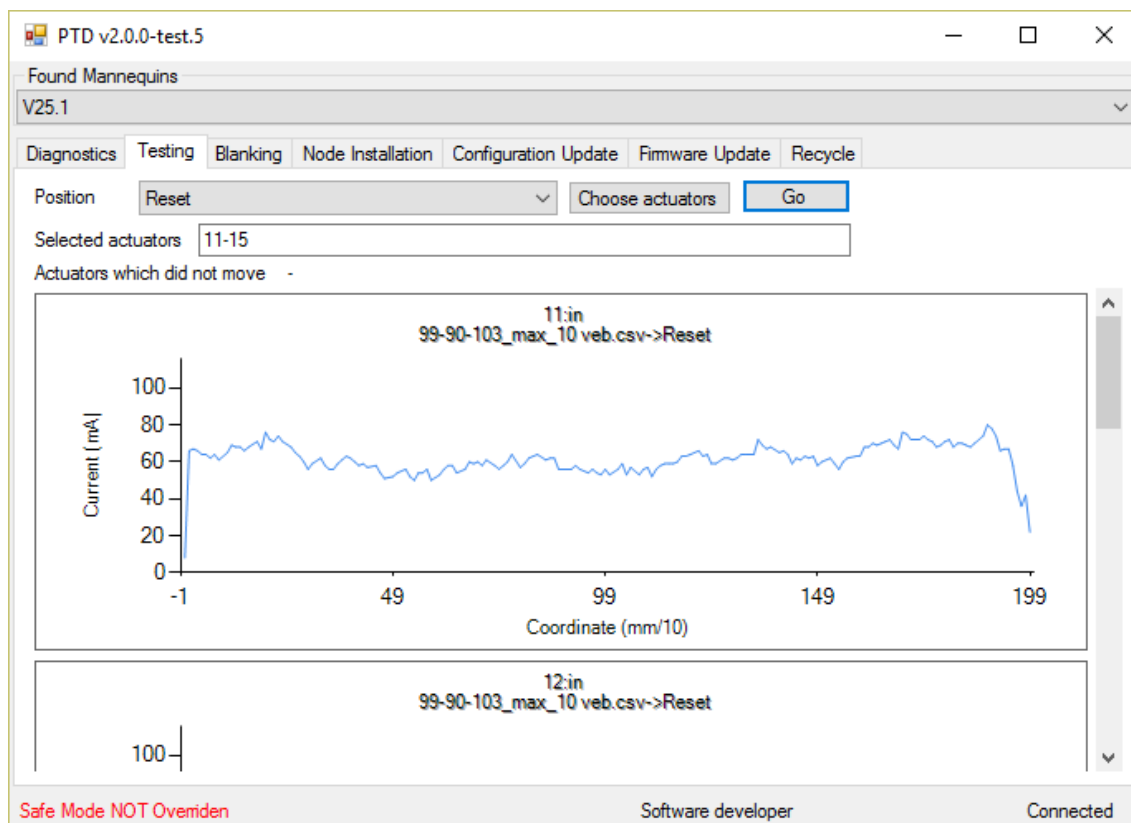
Connected

Joonis 5.2: Sakk *Diagnostics*

kirjet ei leitud *Support Software*'i andmebaasist. Juhul, kui mõni aktuaator on veaseisundis, kuvatakse viimase staatust veerus *Error*. Võimalikud aktuaatori juures esinevad veaseisundid on toodud järgnevalt.

- *Safe mode* - üldine veaolukord.
- *Stall* - tarbitav vool on liiga kõrge.
- *Timeout* - liikumine nõutud pikkuseks võttis liiga kaua aega.
- *Edge tolerance* - koduanduri aktiveerumisel on aktuaatori pikkus liiga palju erinev algsest pikkusest.
- *Configuration* - EEPROM-ist konfiguratsiooniväärtuste lugemine ebaõnnestus.
- *ST Voltage* - toitepinge on piiridest väljas.
- *ST Homing* - koduanduri testi tulemus oli negatiivne.
- *ST code wheel* - enkooderi testi tulemus oli negatiivne.

Aktuaatorite veaseisundite tuvastamisel kasutatakse iga aktuaatori tüübi jaoks *Support Software*'i andmebaasis defineeritud parameetreid (näiteks maksimaalse lubatud tarbitava voolu suurus).

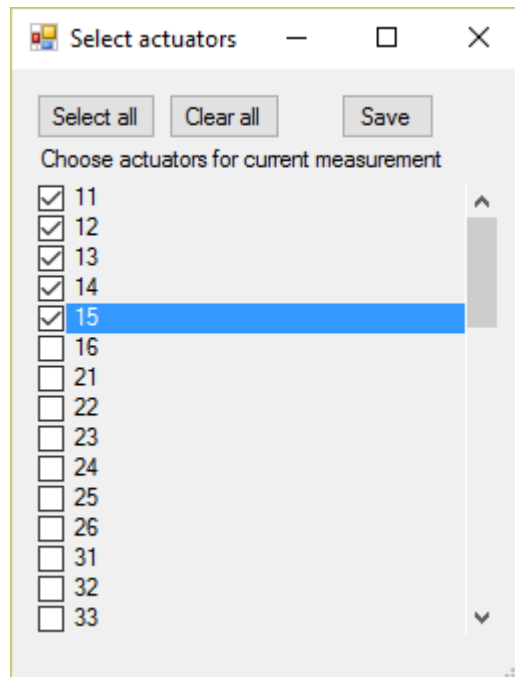


Joonis 5.3: Sakk *Testing*

5.1.3 Sakk *Testing*

Joonisel 5.3 on kujutatud sakk *Testing*. Haldustarkvarasse PTD on juurde lisatud kogu saki funktsionaalsus, sest PTD algversioonis see puudus. Antud saki alla on koondatud aktuaatorite liikumise testimise ja voolutarbe info kogumine. Liikumist saab testida ning voolutarvet mõõta, kasutades pildistamisasendeid või liigutades kõiki aktuaatoreid ühele koordinaadile. Liikumise valik on realiseeritud rippmenüüga, kus saab valida pildistamisasendite ja koordinaadile liikumiste vahel. Asendite info loetakse *Support Software*'i andmebaasist. *Choose actuators* nuppu vajutades on võimalik valida, kas infot soovitakse ühe, mitme konkreetse aktuaatori või kõigi aktuaatorite kohta. Joonisel 5.4 on kujutatud aktuaatorite valiku aken. Nuppu *Go* vajutades võtab mannekeen sisse valitud asendi ning kuvab määratud aktuaatorite voolugraafikud. Aktuaatoreid on võimalik valida ka kasutades *Selected actuators* tekstikasti, kuhu saab sisestada aktuaatorite identifikaatorid üksikhaaval või vahemike kaupa. Vahemikud määratakse kujul *Esimese aktuaatori identifikaator-viimase aktuaatori identifikaator* ning gruppide eraldamiseks on koma.

Iga aktuaatori kohta kuvatakse eraldi graafik, mille x-teljel on koordinaat (aktuaatori liikumise pikkus kümnendikmillimeetrites) ning y-teljel voolutarve. Graafikutel kuvatud info paremaks võrdlemiseks on kõikide graafikute y-telgede maksimaalsed väärtused samad. Kui näiteks ühel aktuaatori maksimaalne voolutarve liikumisel on 100 mA ning teisel 50 mA, siis saab kasutaja graafikult vaadates aru, et teise aktuaatori poolt tarbitav vool on umbes kaks korda väiksem esimese aktuaatori poolt tarbitavast voolust.



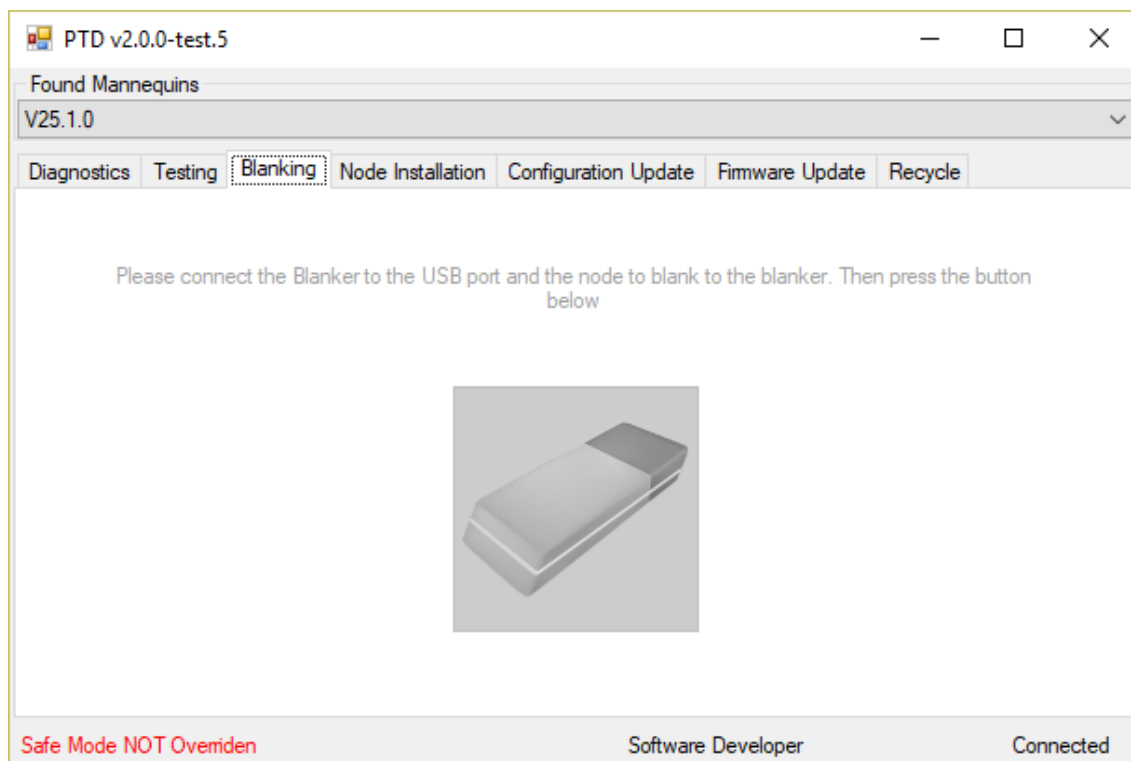
Joonis 5.4: Aktuaatorite valimine voolugraafikute kuvamiseks

Graafiku päises kuvatakse aktuaatori identifikaator ning liikumise suund (sisse või välja). Selle all on info, milliste asendite vahel liiguti (*eelmine asend* → *praegune asend*). Eelmise asendi infot kuvatakse alates teisest liigutamisest, sest esimesel liigutamisel ei ole teada, millises kehakuju asendis mannekeen on.

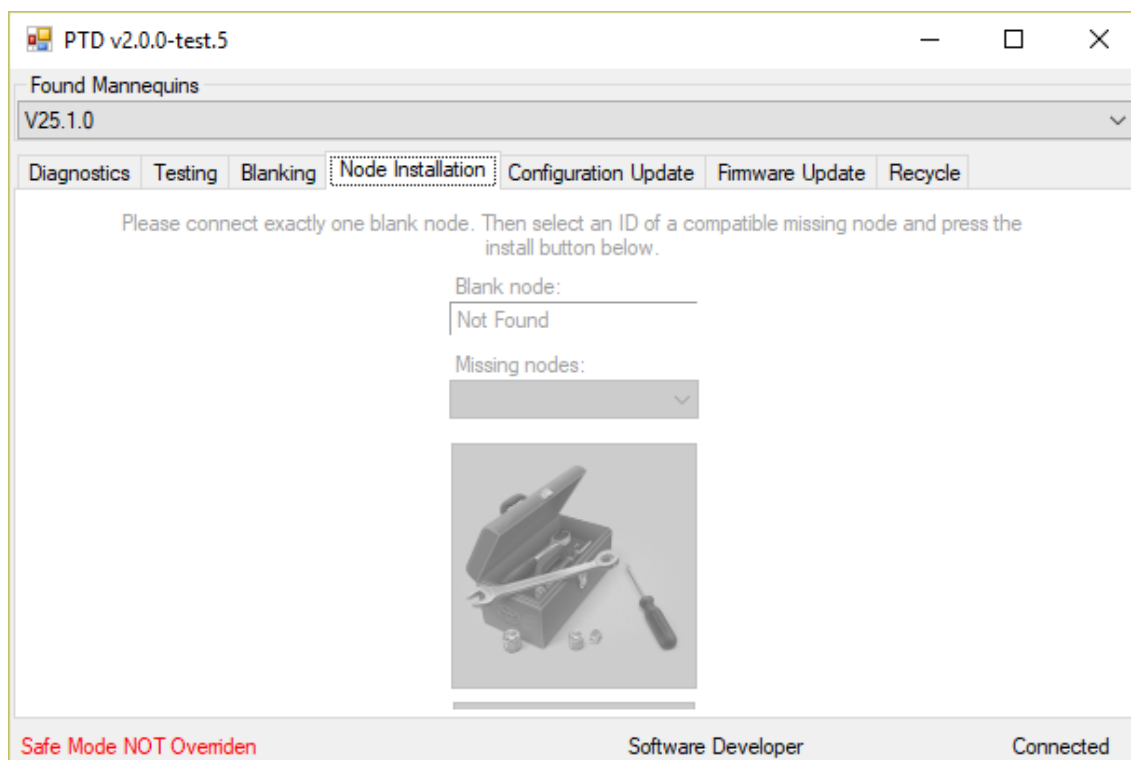
5.1.4 Sakk *Blanking*

Joonisel 5.5 on kujutatud sakk *Blanking*. Antud saki alla on koondatud aktuaatori konfiguratsiooniväärtuste kustutamisega seotud tegevused. Väärtuste kustutamine võimaldab mannekeenil olnud aktuaatorit soovi korral kasutada uue mannekeeni küljes. Aktuaatori konfiguratsiooniväärtused saab kustutada, vajutades kustutuskummi pildiga nuppu.

Kustutamisel eemaldatakse andmebaasis aktuaatori kirjelt seos mannekeeniga ning uuendatakse aktuaatori identifikaatori väärtust. Lisaks logitakse tegevuse sooritanud kasutaja nimi, kuupäev ja kellaaeg. Protsessi õnnestumisel kuvatakse kasutajale vastavasisuline info. Kui konfiguratsiooniväärtuste kustutamine peaks ebaõnnestuma, siis küsitakse kasutajalt vabas vormis kommentaari tegevuse ja vea kohta (näiteks "testisin uut haldustarkvara versiooni ning kustutamine ebaõnnestus"). Kasutaja kommentaar seotakse logiga ning kirjutatakse andmebaasi koos täpsema infoga, millise alamtegevuse juures viga esines. Haldustarkvara algversioonis ei logitud tegevusi ega veainfot andmebaasi ning see on lisatud, et saada ülevaade mannekeenide ja aktuaatoritega tehtud tegevustest ning esinevatest vigadest.



Joonis 5.5: Sakk *Blanking*



Joonis 5.6: Sakk *Node Installation*

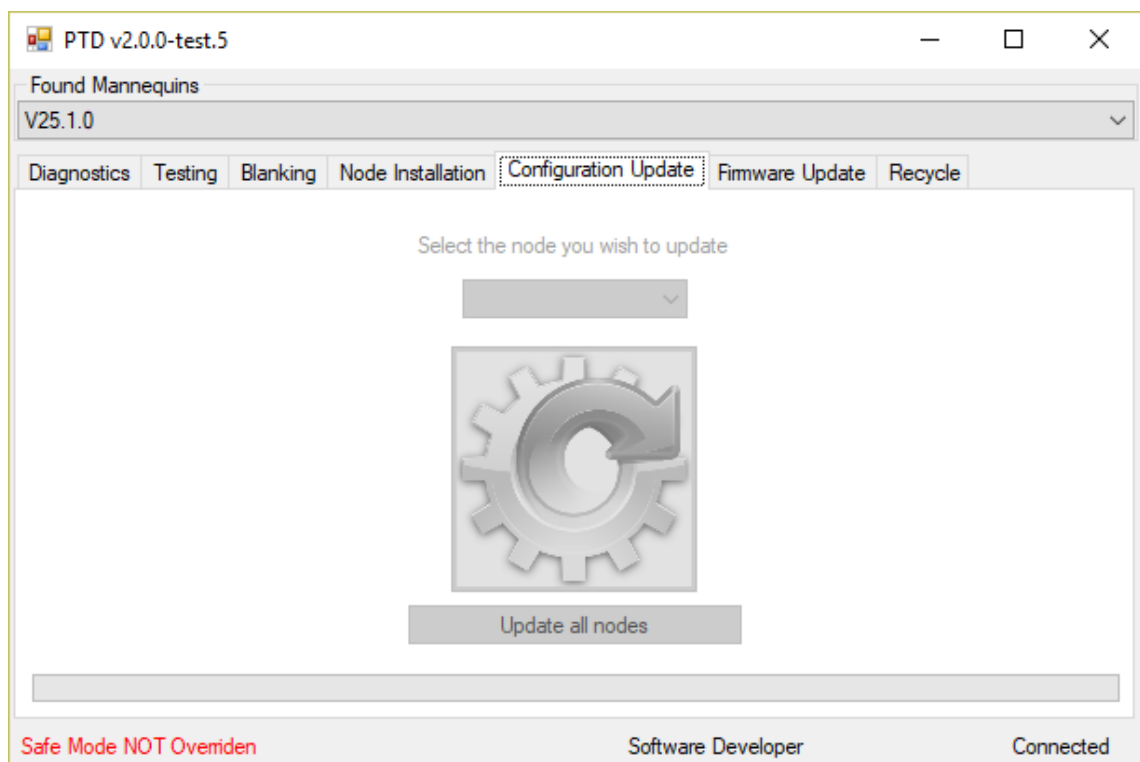
5.1.5 Sakk *Node Installation*

Joonisel 5.6 on kujutatud sakk *Node Installation*. Antud saki alla on koondatud uue aktuaatori mannekeeniga sidumise tegevused. Seda funktsionaalsust kasutatakse

hooldustööde käigus üksikute aktuaatorite vahetamisel ja uue mannekeeni koostamisel kõigi aktuaatorite ühendamisel. Mannekeeni koostamisel lisatakse mannekeenile aktuaatoreid ükshaaval. Kui uus või puhastatud aktuaator on mannekeeniga ühendatud, kuvatakse selle info tekstikastis *Blank node* ning rippmenüüst saab valida, milline identifikaator ning sellel vastavad konfiguratsiooniväärtused aktuaatori EEPROM-i kirjutatakse. Väärtuste kirjutamise käivitab suur mutrivõtmega nupp.

Tegevuse õnnestumisel kuvatakse kasutajale vastavasisuline teade ning mannekeeniga võib ühendada uue aktuaatori. Kui väärtuste kirjutamisel peaks viga tekkima, siis logitakse andmebaasi mannekeeni, aktuaatori, vea tekkepõhjuse ning kasutaja info. Kasutaja saab soovi korral lisada vabas vormis kommentaari tegevuse ja vea kohta, mis lisatakse logile ning kirjutatakse andmebaasi.

5.1.6 Sakk *Configuration Update*



Joonis 5.7: Sakk *Configuration Update*

Joonisel 5.7 on kujutatud sakk *Configuration Update*. Antud saki alla on koondatud aktuaatori konfiguratsiooniväärtuste uuendamise tegevused. Haldustarkvara algversioonis oli võimalik aktuaatoreid uuendada ainult ühekaupa ning lisaks konfiguratsiooniväärtustele uuendati alati ka püsivara. Uues PTD versioonis on püsivara ja konfiguratsiooniväärtuste uuendamine eraldatud.

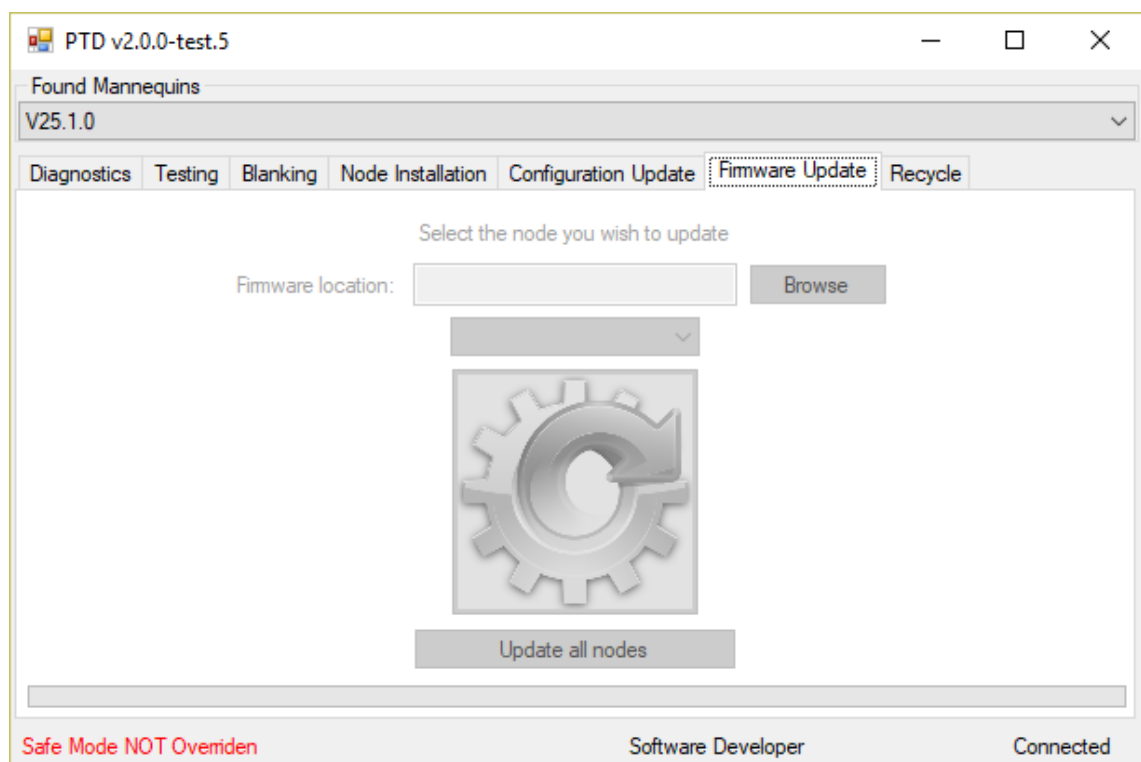
Uuendatava aktuaatori saab valida rippmenüüst ning tegevuse käivitab suur hammasratta ja noole pildiga nupp. Saki alla on lisatud kõikide aktuaatorite konfiguratsiooniväärtuste korraga uuendamise funktsionaalsus ning eemaldatud püsivara uuendamise funktsionaalsus. Kõikide aktuaatorite uuendamiseks on nupp *Update all nodes*.

Uuendamise ajal kuvatakse haldustarkvara allosas progressiriba, mis võimaldab hinnata, kaugel uuendamise järg on.

Uuendamisel logitakse andmebaasi kuupäev, kasutaja nimi ja mannekeeni ning aktuaatori info. Tegevuse õnnestumisel kuvatakse kasutajale vastavasisuline teade. Kui uuendamise protsess ebaõnnestub, siis küsitakse kasutajalt vabas vormis kommentaari tegevuse ja vea kohta, seejärel kirjutatakse veainfo andmebaasi ning seotakse uuendamise logiga. Lisaks salvestatakse täpsem info selle kohta, millise alamtegevuse juures viga esines.

Vea tekkimisel kõikide aktuaatorite uuendamise protsessis katkestatakse uuendamine ning lisaks vea logimisele andmebaasi antakse kasutajale teada, millise aktuaatori konfiguratsiooniväärtuste uuendamine ebaõnnestus. Mannekeeni aktuaatorid on küll poolikult uuendatud, aga väärtuste uuendamise protsessi saab uuesti käivitada. Kui viga tekib ühe aktuaatori juures mitu korda, siis on tarvis see välja vahetada.

5.1.7 Sakk *Firmware Update*



Joonis 5.8: Sakk *Firmware Update*

Joonisel 5.8 on kujutatud sakk *Firmware Update*. Antud saki alla on koondatud aktuaatori püsivara uuendamisega seotud tegevused. Haldustarkvarasse on juurde lisatud kogu saki funktsionaalsust, kuna see puudus PTD algversioonis. Püsivara saab uuendada valitud aktuaatoritel ühekaupa või kõigil aktuaatoritel järjest. Kõigi mannekeeni küljes olevate aktuaatorite identifikaatorid on kirjas rippmenüüs, millest saab valida aktuaatori püsivara uuendamiseks. Üksiku aktuaatori uuendamiseks on suur hammasratta ja noolega nupp. Kõikide aktuaatorite uuendamiseks on nupp *Update all nodes*. Vaikimisi uuendatakse püsivara kõige uuemale versioonile. Kui aktuaatoris on juba kõige uuem püsivara, siis

üksiku aktuaatori uuendamisel antakse sellest kasutajale märku ning uuendamise nupp pole aktiivne. Kõikide aktuaatorite uuendamisel kirjutatakse uus püsivara ainult nendesse aktuaatoritesse, mis vajavad uuendamist. Kui kõik aktuaatorid on uuendatud, ei ole *Update all nodes* nupp aktiivne. Määrates püsivara faili asukoha tekstikastis *Firmware location*, kasutatakse uuendamiseks selles failis leiduvat püsivara.

Eraldiseisvat püsivara uuendamist algses haldustarkvara versioonis ei olnud. Tarkvara uuendamisel uuendati alati ka konfiguratsiooniparameetrite väärtused, aga nüüd on püsivara ja konfiguratsiooniväärtuste uuendamine üksteisest eraldatud. Uuendamise õnnestumisel kuvatakse kasutajale vastavasisuline info. Vea tekkimisel logitakse andmebaasi mannekeeni, aktuaatori, vea tekkepõhjuse ning kasutaja info. Kasutaja saab soovi korral lisada vabas vormis kommentaari tegevuse ja vea kohta, see lisatakse logile ning kirjutatakse andmebaasi. Vea tekkimisel püsivarade järjest uuendamisel peatatakse protsess ning lisaks vea logimisele andmebaasi antakse kasutajale teada, millise aktuaatori uuendamine ebaõnnestus.

Püsivara uuendamise protsessis on oluline, et andmebaasis oleks ajakohane info aktuaatori püsivara versiooni kohta. Juhul kui püsivara kirjutamine aktuaatorisse õnnestub, aga andmebaasis olevaid väärtuseid ei õnnestu muuta, siis taastatakse aktuaatoris olnud püsivara versioon.

5.1.8 Sakk *Recycle*

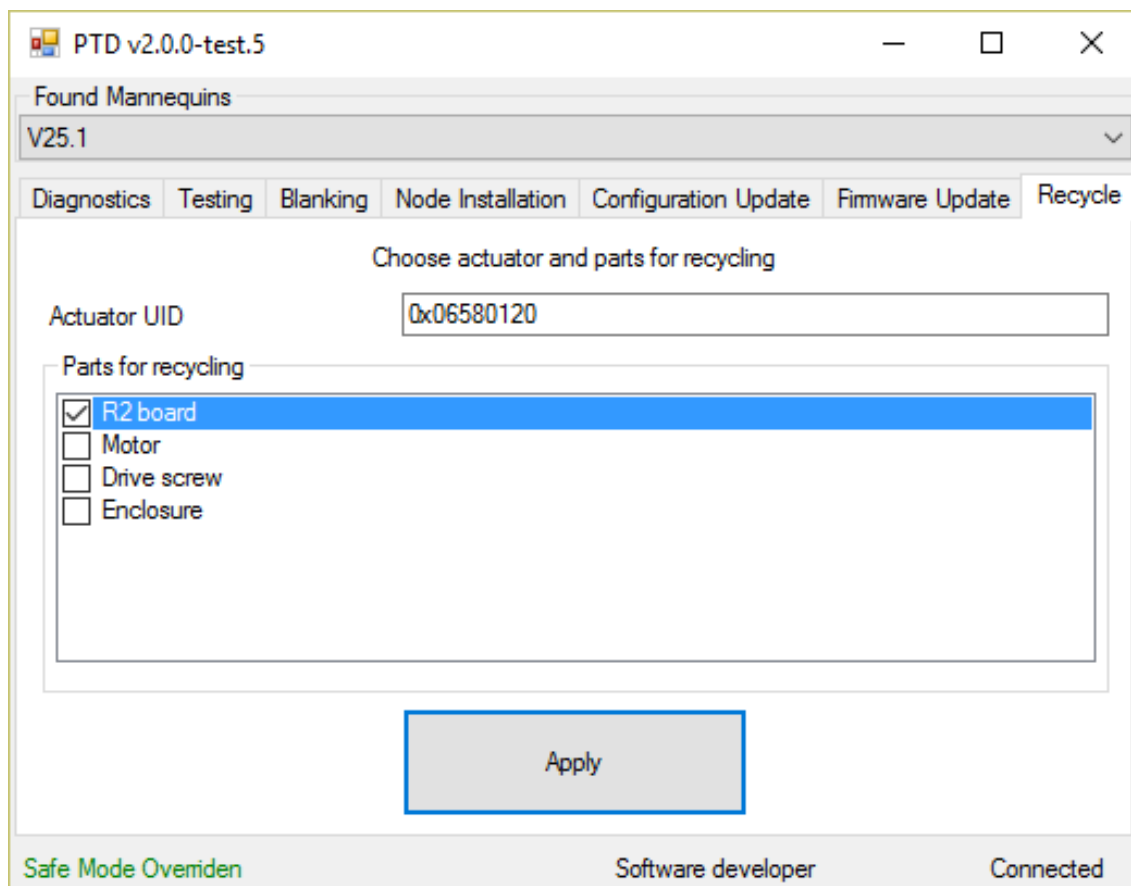
Joonisel 5.9 on kujutatud sakk *Recycle*. Selles sakis on ainult kasutajaliidese elemendid, mis on tehtud tuleviku tarbeks. Saki alla on planeeritud koondada võimalus taaskasutada aktuaatorite komponente. Funktsionaalsust saki alla lisatud ei ole, kuna mannekeeni komponentide laosüsteemi arendamist ei ole veel alustatud.

Väli *Actuator UID* määrab, millise aktuaatori komponente soovitakse taaskasutada ning allpool saab mitmikvalikut kasutades märkida taaskasutatavad osad. Aktuaatori UID lugemise ebaõnnestumisel saab selle sisestada käsitsi.

5.1.9 Parameetrite kasutamine haldustarkvara käivitamisel

Andmete hoidmise jaoks on kasutuses kaks erinevat andmebaasi. Ühes hoitakse testandmeid ning seda kasutatakse tarkvara arendamisel. Teises hoitakse andmeid, mida kasutatakse töös mannekeenidega kõigis mannekeeni elutsükli etappides. Parameetriga **-debug** saab valida kumba andmebaasi PTD kasutab. Valitud andmebaasi kasutatakse kõikide toimingute jaoks, mis vajavad andmebaasi. Näiteks käivitades programmi käsuga `ptd.exe -debug` kasutatakse testandmebaasi.

Vea tekkimisel kirjutatakse vea kirjeldus andmebaasi. PTD arendamise ja testimise käigus on oluline, et oleks võimalus veainfo andmebaasi kirjutamine vajadusel mugavalt välja lülitada, et andmebaasi mitte üle koormata üleliigsete, korduvate ja ebavajalike kirjetega. Käivitades programmi käsuga `ptd.exe -noreport` ei kirjutata veainfot andmebaasi.



Joonis 5.9: Sakk *Recycle*

5.2 Mannekeeni tarkvara rakendusliides

Rakendusliidesesse oli vaja lisada kogu uus funktsionaalsus, mida PTD vajab mannekeeniga suhtlemiseks. CAN-pakettide saatmiseks ja vastuvõtmiseks rakendusliideses on vajalik kõigepealt lisada pakettide kirjeldused (paketi identifikaator ning sisu andmetüüp ning pikkus baitides). Aktuaatorite voolutarbimise info pärimiseks on lisatud voolu, liikumise lõpu ning saada voolu pakettide kirjeldused. Lisatud on ka funktsioonid, mis võimaldavad pakette rakendusliidesest sillale saata ning mis annavad märku, kui pakett on sillalt saabunud.

Samaaegselt olid kasutuses mitme erineva riistvaraga mannekeenid, seega oli arendamise käigus oluline, et rakendusliidesesse lisatud funktsionaalsus töötaks ka vanemate silla ja aktuaatori püsivara versioonidega. Vooluinfo küsimiseks on olemasolevale liikumise käsule lisatud üks bait, mis määrab, kui tihti vooluinfot saadetakse. Vanemates tarkvaraversioonides, kus voolupakette implementeeritud ei ole, ignoreeritakse lõpus olevat baiti. Kui bait on 0, siis infot ei saadeta ning kui nullist suurem, siis arv tähistab saatmise sagedust. Näiteks kui number on 4, siis infot saadetakse iga liigutud 0.4 mm kohta. Seega arv näitab, mitme liigutud kümnendikmillimeetri kohta infot saadetakse.

5.3 *Support Software*'i rakendusliides

Andmebaasiga ühendumiseks oli varasemalt kasutuses andmebaasi rakendusliides. *Support Software*'i uuendamisega vahetus andmebaas ning andmebaasi rakendusliidest ei olnud enam mõistlik kasutada. Vanas rakendusliideses puudusid kasutajakontod ning logimisinfo oli kirjutatud rakendusliidesesse endasse. Seetõttu oleks igauks, kellel oli ligipääs tarkvarale, mis kasutas andmebaasi rakendusliidest, saanud andmebaasis oma tahtmise järgi andmeid lugeda ja muuta.

Support Software'i rakendusliides on kirjutatud keeles C# ning ühendub andmebaasiga kasutades *Support Software*'i. Andmevahetus toimub üle HTTPS-protokolli, kasutades GET, PUT, POST ja DELETE meetodeid. Seda liidest varasemalt ei olnud ning luua oli vaja kogu funktsionaalsus.

Haldustarkvara jaoks oli vaja lisada meetodid, mis võimaldavad mannekeenide ja aktuaatorite infot lugeda ja muuta ning mannekeeni konfiguratsiooniinfot lugeda. Lisaks oli vaja lisada funktsionaalsus kasutajainfo lugemiseks ning kasutajate autentimine, püsivara allalaadimine ning veainfo ja tegevuste ajaloo kirjutamine.

5.4 Veebirakendus *Support Software*

Uue haldustarkvaraga paralleelselt toimus uue *Support Software*'i arendamine. Selles veebirakenduses hoitakse mannekeenide ja aktuaatorite infot, mida saab vaadata, lisada ja muuta. Andmebaasiga ühendumiseks on vajalik *Support Software*'is autoriseerimine ning pärast seda saab andmebaasi andmeid kasutada, kasutades HTTP-päringuid. Mugavamaks ühendumiseks on loodud *Support Software*'i rakendusliides, mis asendab andmebaasi rakendusliidest. Veebirakenduse vastused päringutele on JSON-formaadis.

Haldustarkvara tarvis on veebirakendusse lisatud järgnev funktsionaalsus.

- Kasutajarollide ja õiguste lisamine, kasutajarollidele õiguste määramine ning kasutaja rolli lugemine.
- Mannekeeni konfiguratsiooni ning aktuaatorite info lugemine.
- Mannekeeniga seotud aktuaatorite info lugemine ja muutmine.
- Aktuaatori püsivara allalaadimine.
- Haldustarkvara logi kuvamine ja lisamine.

5.5 Aktuaatori püsivara

Aktuaatori liikumise voolutarbe info kasutamiseks on voolu ja koordinaadi infot vaja saata kindla vahemaa järel. Aktuaatori püsivara algversioonis saadeti ainult üks väärtus - maksimaalne voolutarbe näit kogu liikumise ulatuses, mis ei andnud piisavalt täpset infot aktuaatori kohta.

Lisatud on pakett, mida aktuaator saadab, kui vooluinfot on küsitud:

- hetkel tarbitav vool milliamprites (2 baiti) ning
- aktuaatori koordinaat kümnendikmillimeetrites (2 baiti).

Eemaldatud on liikumise lõpus vooluinfo saatmine, kuna see ei olnud enam vajalik. Lisatud on pakett aktuaatori liikumise lõpetamisest teatamiseks. See pakett oli vajalik, et teada saada kui aktuaatori liikumine on lõppenud ja saabunud andmed on kompleksed, ning informeerimiseks, kui aktuaator ei ole mingil põhjusel ette antud koordinaadile jõudnud.

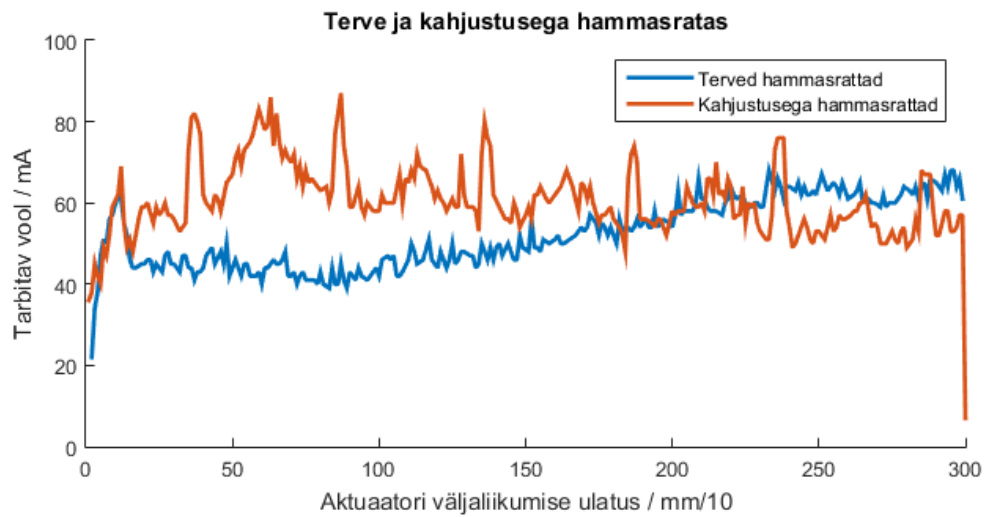
6 Aktuaatori voolutarbimise analüüs

Mannekeeni ülesanne on võtta sisse erinevaid kehakujusid ning kogu mannekeeni liikumine on üles ehitatud aktuaatoritele. Seetõttu on aktuaatorite töökindluse ja vastupidavuse tagamine mannekeenide juures väga oluline. Aktuaatorite voolutarbe info kasutatavuse näidisenä on toodud tulemused testmannekeenilt V25.1. Aktuaatorite voolud on mõõdetud liikumisel algasendist 5 mm, 15 mm ja 30 mm kaugusele (30 mm juures olid kõik aktuaatorid jõudnud maksimaalse lubatud käigupiiranguni). Järgnevalt on leitud iga aktuaatori keskmine voolutarve. Mannekeenide ja aktuaatorite disainimisel on määratud, et aktuaatorite voolutarve peaks jääma vahemikku 40 ± 10 mA. Lisas 2 on toodud mannekeeni V25.1 aktuaatorite keskmised voolutarbed.

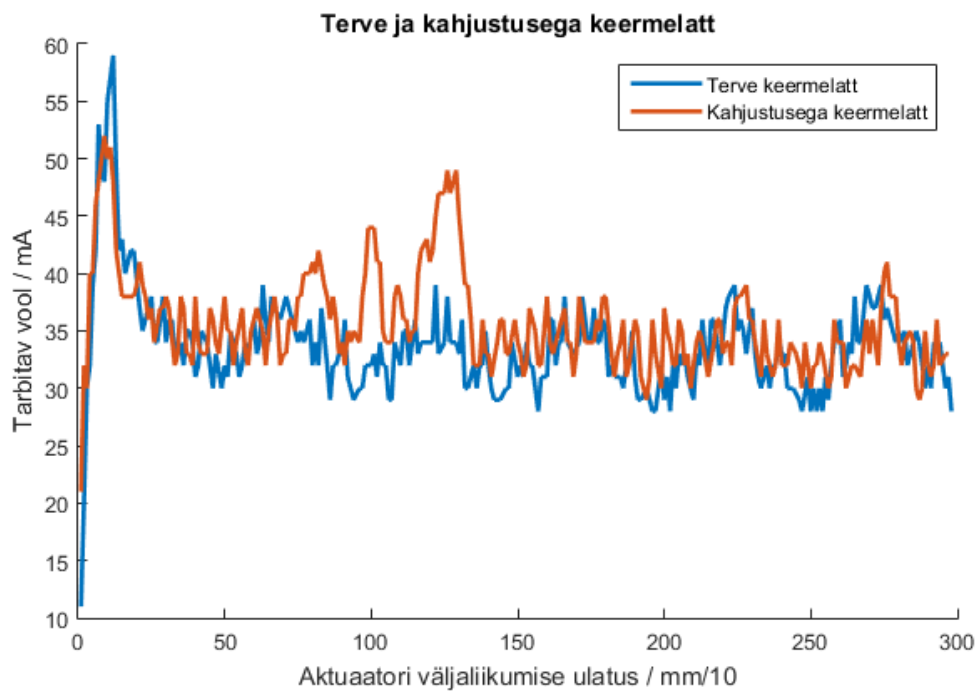
Tabelist 8.2 (lisas 2) on näha, et kuueteistkümne aktuaatori voolutarve on etteseadud piiridest väljas. See tähendab $16/50 = 32\%$ V25.1 aktuaatoritest ei vasta disainis etteantud nõuetele. Võrreldes etteseadud voolupiirangut histogrammiga, mis on kujutatud lisas 2, joonisel 8.1, on näha, et algselt disaini käigus määratud piirid ei pruugi sobivad olla. Samas on histogrammilt näha, et osade aktuaatorite keskmine voolutarve on oluliselt suurem kõikide V25.1 aktuaatorite voolutarvete keskmisest. See annab märku, et vastavaid aktuaatoreid on vaja järgmise hoolduse käigus täpsemalt kontrollida. Keskmisest oluliselt suurema voolutarbe põhjus on arvatavasti aktuaatoris endas või mannekeeni konstruktsioonis. Mõlemal juhul lüheneb aktuaatori tööiga ning seega väheneb ka mannekeeni töökindlus.

Joonisel 6.1 on kujutatud aktuaatori 21 voolutarve. Graafikult on näha, et voolu tarbimine on perioodiliselt kõrgem. Põhjuse leidmiseks sai aktuaator lahti võetud ning mehaanikat uurides selgus, et hammasratas oli saanud kahjustada. Kahjustus on kuvatud lisas 3, joonisel 8.2. Antud aktuaatoris on kasutusel 5 mm sammuga keermelatt ning voolutarbe graafikult on näha, et vool kasvab järsult iga 5 mm tagant. Antud juhul on selge, et voolutarbimise graafikult nähtud andmete põhjal on käesolev viga ennetatav. Tulemusena paraneb mannekeenide töökindlus, sest vea saab tuvastada ja parandada enne kui vigane aktuaator töötamise lõpetab.

Joonisel 6.2 on kujutatud aktuaatori 84 voolutarve. Graafikult on näha, et voolutarve on suurem vahemikus 7.5 mm - 12.5 mm. Aktuaator sai põhjuse leidmiseks lahti võetud ning näha oli, et keermelatt oli kahjustusega. Kahjustus on kujutatud lisas 3, joonisel 8.3. Antud juhul on samuti voolutarbimise graafikult nähtud andmete põhjal viga leitav ja ennetatav. Isegi kui voolugraafikult ei ole võimalik täpset viga välja lugeda, annab see piisavalt infot, et ennetada olukorda, kus aktuaator lõpetab vea tõttu töö. Seega haldustarkvaras voolutarbe kuvamine võimaldab juba regulaarse hoolduse käigus identifitseerida ja välja vahetada kulunud või defektseid aktuaatoreid ning seeläbi ennetada tööseisakuid mannekeenide kasutamisel.



Joonis 6.1: Aktuaatori 21 voolutarve (punane) ning terve hammasrattaga aktuaatori voolutarve (sinine)



Joonis 6.2: Aktuaatori 84 voolutarve (punane) ning terve keermelatiga aktuaatori voolutarve (sinine)

7 Robotmannekeenide haldustarkvara testimise plaan

Haldustarkvara testimise eesmärgiks on tagada PTD vastavus nõuetele ning ka korrektne töötamine, kui haldustarkvarasse või sellega seotud tarkvarakomponentidesse muudatusi tehakse. Järgnevalt on kirjeldatud tulevikus tehtavate arenduste tarbeks koostatud funktsionaalse testimise plaan. Testid on vaja sooritada kui haldustarkvara või sellega seotud komponentide funktsionaalsust muudetakse. Testid on kirjeldatud sellise täpsusega, et iga mannekeeni arendamisega seotud meeskonnaliige oleks võimeline neid läbi viima.

Testid jagunevad kaheksaks osaks (alapeatükid 7.1 - 7.8). Iga testi puhul on oluline läbi teha kõik sammud ning kõik tõrked dokumenteerida. Testide 7.5-7.8 juures peab täiendavalt veenduma, et neid tegevusi saaks sooritada ainult siis, kui mannekeenil olev mehaaniline lukk on suletud asendis. Kõik testid kasutavad testandmebaasis olevaid väärtuseid. Viimase kasutamiseks peab PTD käivitama lipuga `-debug` ning valima, et kasutatakse testandmebaasi.

7.1 Üldtest

Nende testide eesmärgiks kontrollida haldustarkvara baasfunktsionaalsust: mannekeeni ja andmebaasiga ühendumine.

- Mannekeeni ühendades peab rippmenüüsse tekkima uue mannekeeni kirje ning kui mannekeen ühendatakse arvuti küljest lahti, siis kirjet ei tohi enam rippmenüüs olla.
- Keerates mannekeenil oleva luku suletud asendisse, peab toimuma ohurežiimi ignoreerimine. Sellest annab teada PTD alumises vasakus nurgas olev infotekst.
- Kui ühendus andmebaasiga katkeb, peab alumises paremas nurgas olev infokast kuvama *Not connected* ning kui ühendus andmebaasiga on olemas, siis *Connected*. Testimiseks keelata võrguseade või eemaldada võrgukaabel.

7.2 Kasutajarollid ja sisselogimine

Nende testide eesmärgiks on kontrollida, kas kasutaja autentimine ning õiguste piiramine toimib korrektselt.

- Logida vaheldumisi sisse erinevate testkasutajatega. Süsteemi testimiseks on loodud eri rolle kandvad testkasutajad.
- Kontrollida lubatud tegevuste vastavust lisas 1, tabelis 8.1 kirjeldatuga.

7.3 Sakk *Diagnostics*

Nende testide eesmärgiks on kontrollida aktuaatorite info kuvamise funktsionaalsust.

- Kuvatud peavad olema kõik ühendatud aktuaatorid. Kui mõni on puudu, siis kuvatakse *diagnose* veerus vastav info. Testimiseks eemaldada 2-5 aktuaatorit ning kontrollida, et eemaldatud aktuaatorite *diagnose* veerus oleks *MISSING*.
- *Error* veeru testimiseks on loodud komplekt testaktuaatoreid, kus iga aktuaator saadab kindlat veateadet, kui see mannekeeniga ühendada. Tekitatava vea tüüp on neile peale märgitud. Ühendada kõik komplektis olevad aktuaatorid mannekeeniga ning kontrollida, kas haldustarkvaras kuvatakse korrektne veainfo.

7.4 Sakk *Testing*

Nende testide eesmärgiks on kontrollida aktuaatorite voolutarbe mõõtmise ja kuvamise funktsionaalsust.

- Ühendada arvutiga kaks mannekeeni, millest ühel on andmebaasis asendid olemas. Kui asendid puuduvad, kuvatakse ainult *Reset* asend, vastasel juhul kõik, mis on nähtavad ka *Support Software*'ist, mannekeeni asendite alt. Kontrollida on vaja lisaks asendite olemasolule ka nende õigsust (näiteks, et ei oleks kuvatud vale mannekeeni asendeid).
- Eemaldada aktuaator ning üritada asendit võtta. Ilmuma peab veateade.
- Valida voolude mõõtmiseks aktuaatoreid kasutades nii tekstikasti kui ka märkeruutudega valikut. Pärast märkeruutude valiku tegemist ja salvestamist peab tekstikastis olev info uuenema.
- Voolutarbimise info graafikute testimiseks valida umbes kümme aktuaatorit ning selline kehakuju asend, kus valitud aktuaatoritest umbes pooled ei liigu. Kui kehakuju asend on sisse võetud, siis liikunud aktuaatorite kohta kuvatakse voolugraafikud ning ülejäänute kohta info, et need ei liikunud.

7.5 Sakk *Blanking*

Nende testide eesmärgiks on kontrollida aktuaatori konfiguratsiooniväärtuste kustutamise funktsionaalsust.

- Ühendada mini-mannekeen ning aktuaator. Puhastamise nupp peab olema deaktiveeritud.
- Ühendada *blanker* ning aktuaator, mis on mannekeeniga seotud. Kui aktuaatori identifikaator jääb vahemikku 1-1020, peab puhastamise nupp olema aktiivne.
- Kui puhastamine õnnestub, peab aktuaatori identifikaator muutuma 1021-ks. Järgnevalt tuleb *Support Software*'is veenduda, et aktuaatori seos mannekeeniga on eemaldatud.

7.6 Sakk *Node installation*

Nende testide eesmärgiks on kontrollida uue või puhastatud aktuaatori mannekeeniga sidumise funktsionaalsust.

- Ühendada mannekeeniga üks aktuaator, mille identifikaator on 1021. Uue identifikaatori valiku rippmenüü ning kinnitamise nupp peavad olema aktiivsed.
- Ühendada mannekeeniga rohkem kui üks puhastatud (identifikaatoriga 1021) aktuaator ja kontrollida, et kinnitamise nupp ei oleks aktiivne.
- Ühendada mannekeeniga suvaline aktuaator, mille identifikaator ei ole 1021 ning kontrollida, et kinnitamise nupp ei oleks aktiivne.
- Pärast edukat uue aktuaatori mannekeeniga sidumist kontrollida, et aktuaatori identifikaator on muudetud ning mannekeeniga sidumise info on *Support Software*'is.

7.7 Sakk *Configuration update*

Nende testide eesmärgiks on kontrollida aktuaatori konfiguratsiooniväärtuste uuendamise funktsionaalsust.

- Muuta 2-3 aktuaatoril käigupiirangut väiksemaks ning uuendada nende konfiguratsioon ükshaaval. *Testing* sakis võtta maksimaalne asend ning kontrollida, et uuendatud aktuaatorid ei liigu kaugemale uuest käigupiirangust.
- Taastada aktuaatorite algsed käigupiirangud ning uuendada kõikide aktuaatorite konfiguratsioon. Kontrollida sama moodi nagu eelmises punktis.

7.8 Sakk *Firmware update*

Nende testide eesmärgiks on kontrollida aktuaatori püsivara uuendamise funktsionaalsust.

- Võtta kolm-neli aktuaatorit ning kirjutada nende püsivara üle, kasutades vanemat püsivara versiooni. Kontrollida, et sakis *Diagnostics* on püsivara versiooni info uuenenud.
- Uuendada ühe aktuaatori püsivara, kasutades kõige uuemat andmebaasis leiduvat versiooni ning kontrollida *Diagnostics* sakist, et püsivara versiooni info oleks uuenenud.
- Uuendada ülejäänud aktuaatorite püsivara korraga (*Update all nodes*) ning kontrollida, et sakis *Diagnostics* oleks püsivara versiooni info uuenenud.

8 Tulemused ja edasine arendus

Käesoleva magistritöö raames valmis mannekeenide haldustarkvara, mis vastas peaaegu kõikidele esitatud nõuetele. Aktuaatorite taaskasutamisest on implementeeritud ainult kasutajaliides, sest varuosade laosüsteemi nõuded ei olnud töö valmimise ajaks kinnitatud. Lisaks ei teavita haldustarkvara uue versiooni olemasolust, sest tarkvarauuendamise süsteemi ei pidanud kasutama ainult PTD ning selle nõuded ei olnud magistritöö kirjutamise ajaks määratud.

Olulise uuendusena saab valminud haldustarkvara versioonis kasutaja voolutarbograafikute põhjal hinnata, iga aktuaatori tehnilist seisukorda. Antud funktsionaalsust saab kasutada, et tulevikus lisada graafikute automaatne analüüsimine, kus iga aktuaatori kohta kuvatakse potentsiaalsed vead. Vigade tuvastamiseks saab näiteks võrrelda teadaoleva veaga aktuaatorite voolugraafikuid testitavate aktuaatorite voolugraafikutega ning hinnata, kui sarnased andmed on. Teiseks võimaluseks oleks voolutarbimiste graafikult otsida kindlaid elemente, näiteks perioodilisus või impulss(id).

PTD funktsionaalsust saab tulevikus kasutada ka aktuaatorite vigade ennetamiseks. Selleks on tarvis aktuaatorite poolt tarbitavat voolu saab kindla aja tagant mõõta ning võrrelda sama aktuaatori eelmiste näitudega. Mõõtes paljude aktuaatorite voolusid, saab hinnata kui suure voolutarbimise juures aktuaatorid keskmiselt töötamise lõpetavad. Seda infot kasutades on võimalik aktuaatoreid ennetavalt välja vahetada.

Kokkuvõte

Käesoleva magistritöö eesmärgiks oli mannekeenide haldustarkvara funktsionaalsuse täiendamine ja parandamine. Lõppeesmärgiks oli uue haldustarkvara versiooni valmimine, mis sisaldab nõuetes kirjeldatud funktsionaalsust. Lisaks oli vaja välja töötada funktsionaaltestide plaan, mille abil saab hinnata vastavust nõuetele ning tuvastada nii PTD kui sellega seotud komponentide edasisel arendamisel kui osa olemasolevast funktsionaalsusest enam ei tööta.

Esiteks tuli kirjeldada tarkvarale esitatud nõuded. Peale seda oli vaja määrata, kui suures osas haldustarkvara algversioon nõutele vastab. Lisaks oli vaja kirjeldada olulisemad kasutusjuhud ning peale seda kirjeldada erinevate kasutajarollide õigused.

Eeltööde ja uuringute peatükk kirjeldas, kuidas luua intuitiivset kasutajaliidest, millised on erinevad tarkvara testimise meetodid ning kuidas saab robotite töö käigus tekkivaid vigu ennustada ja ennetada. Lisaks on selles peatükis kirjeldatud, kuidas tuvastada aktuaatori mehaanika vigasid, kasutades voolutarbimise infot.

Tarkvaraarenduse peatükk kirjeldas, milline funktsionaalsus on haldustarkvarasse lisatud ning kuidas kasutajaliides välja nägi. Lisaks on kirjeldatud, millist funktsionaalsust oli vaja lisada haldustarkvaraga seotud tarkvarakomponentidesse.

Aktuaatori voolutarbimise peatükis uuriti, kas ja kuidas saaks aktuaatorite voolutarbe infot kasutada. Seal võrreldakse vigaseid aktuaatoreid töökorras olevatega, et näha kas vea olemasolu ja tüüpi on võimalik määrata, kasutades aktuaatori voolutarbimise info graafikuid.

Testimise peatükk kirjeldas haldustarkvara testimise plaani. Testide eesmärgiks oli kontrollida PTD funktsionaalsust ning vastavust nõuetele. Tarkvara testid olid kirjeldatud piisava täpsusega, et mannekeeni arendusmeeskonna liikmed oskaksid teste sooritada.

Viimases peatükis kirjeldati, kui suur osa nõutud funktsionaalsusest sai implementeeritud ning mida on vaja teha, et aktuaatorite voolutarbimise infot saaks kasutada vigade ennustamiseks ja mannekeeni hooldusaegade paremaks määramiseks.

Summary

Development of management software and connected components for robotic mannequin.

Antti Jaaniste

The goal of this thesis was to fix old functionality of the mannequin management software and add new one. Software which has the functionality described in the requirements had to be released. In addition to finishing the software development, testing plan, which helps to determine whether software has all the required functionality, was created.

Firstly, requirements for the software are described. Secondly, it was necessary to determine how much of the required functionality was already implemented in PTD. After that, most important use cases and permissions for users are described.

Preliminary work and research chapter describes how to design intuitive user interface and how to predict and prevent faults in robotics systems. The chapter also describes how to detect faults in actuator mechanics using current consumption data.

Software development chapter describes, which functionality was added to PTD and how the user interface looked like. Also the work done on the connected software components is described.

Actuator current consumption analysis chapter describes whether it possible to use current consumption data to determine actuator faults. Working actuators were compared to faulty ones, to see if current consumption graph has any indication of the fault.

Testing chapter describes the created plan for software functional testing. Goal of the tests was to determine whether PTD has all the required functionality. Test descriptions have the necessary information, so all the development team members are capable to run the tests.

Last chapter describes which required functionality was implemented and what future development can be done to use current consumption data for fault prediction.

Kirjandus

- [1] Arduino. EEPROM Library. <https://www.arduino.cc/en/Reference/EEPROM>. 08.05.2017.
- [2] Daniel Penn Associates. Defining Preventive and Predictive Maintenance. <http://www.danielpenn.com/insights-resources/case-studies/preventive-predictive-maintenance/>. 03.05.2017.
- [3] Atmel. JTAGICE3. <http://www.atmel.com/tools/JTAGICE3.aspx>. 05.04.2017.
- [4] Everaldo Coelho and YellowIcon. All Crystal icons were posted by the author as LGPL on kde-look. <https://commons.wikimedia.org/w/index.php?curid=1159346>. 02.05.2017.
- [5] Everaldo Coelho and YellowIcon. All Crystal icons were posted by the author as LGPL on kde-look. <https://commons.wikimedia.org/w/index.php?curid=1160581>. 02.05.2017.
- [6] MicroMain Corporation. What is Preventive Maintenance? <https://www.micromain.com/what-is-preventive-maintenance/>. 03.05.2017.
- [7] Farlex. The free dictionary - UID. <http://acronyms.thefreedictionary.com/uid>. 31.03.2017.
- [8] Introducing JSON. JSON. <http://www.json.org/>. 02.05.2017.
- [9] KEIL. General: Intel HEX file format. <http://www.keil.com/support/docs/1584/>. 08.05.2017.
- [10] Frederic Lefrense. Predictive maintenance: 8 keys to a successful implementation. <http://www.mobility-work.com/blog/articles/predictive-maintenance-8-keys-successful-implementation>. 03.05.2017.
- [11] Evert McKay. Intuitive UI: What the heck is it? <http://www.uxdesignedge.com/2010/06/intuitive-ui-what-the-heck-is-it/>. 02.05.2017.
- [12] Microsoft Developer Network. DataGridView Class. [https://msdn.microsoft.com/en-us/library/system.windows.forms.datagridview\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.datagridview(v=vs.110).aspx). 02.05.2017.
- [13] Microsoft Developer Network. ListView Class. [https://msdn.microsoft.com/en-us/library/system.windows.forms.listview\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.listview(v=vs.110).aspx). 02.05.2017.

- [14] Jared M. Spool. What Makes a Design Seem ‘Intuitive’?. https://articles.uie.com/design_intuitive/. 02.05.2017.
- [15] Software testing fundamentals. Ad Hoc Testing. <http://softwaretestingfundamentals.com/ad-hoc-testing/>. 03.05.2017.
- [16] Software testing fundamentals. Black Box Testing. <http://softwaretestingfundamentals.com/black-box-testing/>. 02.05.2017.
- [17] Software testing fundamentals. Gray Box Testing. <http://softwaretestingfundamentals.com/gray-box-testing/>. 03.05.2017.
- [18] Software testing fundamentals. Software Testing Methods. <http://softwaretestingfundamentals.com/software-testing-methods/>. 02.05.2017.
- [19] Software testing fundamentals. White Box Testing. <http://softwaretestingfundamentals.com/white-box-testing/>. 02.05.2017.
- [20] KDE UserBase. Plasma. <https://userbase.kde.org/Plasma>. 02.05.2017.
- [21] Vallaste. Vallaste e-teatmik. <http://www.vallaste.ee/>. 24.03.2017.

Lisad

Lisa 1. Kasutajarollide õigused haldustarkvaras

Tabel 8.1: Erinevate kasutajarollide õigused

Tegevus	Tarkvara arendaja	Elektroonika arendaja	Mehaanika arendaja	Tehniline tugi	Noorem tehniline tugi
Info vaatamine	+	+	+	+	+
Konfiguratsiooni kustutamine	+	+	+	+	
Konfiguratsiooni kustutamine <i>blankerita</i>	+				
Konfiguratsioon uuendamine kõigile	+	+	+	+	
Konfiguratsioon uuendamine valitutele	+		+		
Püsivara uuendamine	+	+	+	+	
Püsivara vahetamine	+				
Konfiguratsiooni hoiatuste vaatamine	+	+	+	+	+
Aktuaatori taaskasutamine	+	+	+		
Kasutajarollide vahetamine	+				
Aktuaator taaskasutusse	+	+	+		
Aktuaatori läbisõidu hoiatuse vaatamine	+	+	+	+	+

Lisa 2. Aktuaatorite keskmised voolutarbed

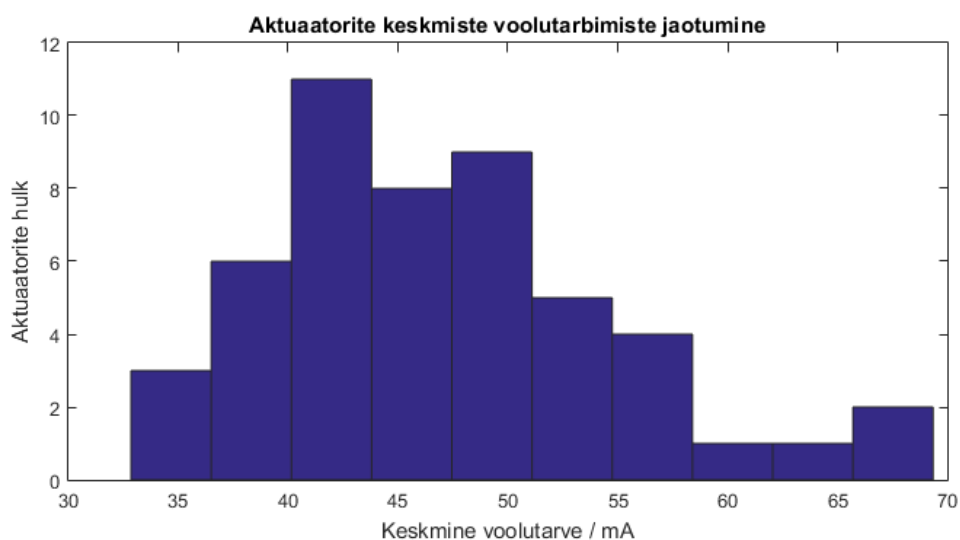
Tabel 8.2: Mannekeeni V25.1 aktuaatorite poolt tarbitavad voolud. Rohelisega on märgitud voolutarve 40 ± 10 mA ning punasega sellest suurem voolutarve.

Aktuaator	Keskmine voolutarbimine liikumisel			
	0 → 5 mm	0 → 15 mm	0 → 30 mm	Liikumiste keskmine
11	64.19	60.92	61.53	62.21
12	37.90	43.42	45.31	42.21
13	41.81	47.94	41.23	43.66
14	47.40	51.86	47.46	48.90
15	40.71	45.42	51.47	45.87
16	40.96	43.52	50.48	44.99
21	50.46	49.67	67.22	55.78
22	34.57	39.51	47.52	40.53
23	33.42	32.68	41.01	35.70
24	63.16	53.23	56.48	57.62
25	45.46	53.24	59.20	52.63
26	65.19	69.41	73.45	69.35
31	34.62	35.55	41.64	37.27
32	42.89	48.33	51.54	47.59
33	49.08	48.44	53.27	50.26
34	46.93	50.82	60.74	52.83
35	37.22	42.88	48.03	42.71
36	51.81	54.32	51.42	52.51
41	49.67	55.22	60.57	55.15
42	37.36	38.55	51.79	42.57
43	43.87	51.62	44.57	46.68
44	40.66	51.10	51.07	47.61
45	53.13	55.63	57.04	55.27
46	48.19	48.48	48.92	48.53
51	46.66	48.65	54.03	49.78
52	36.24	40.79	47.34	41.46
53	47.17	47.69	46.01	46.96
54	41.64	43.33	43.65	42.87
55	55.73	58.84	69.61	61.40
56	38.40	40.84	40.53	39.92
61	50.36	54.18	52.06	52.20
62	42.23	40.83	40.97	41.35
63	42.33	42.34	44.14	42.94
64	33.25	30.62	34.68	32.85
65	38.76	36.99	34.74	36.83
66	38.15	38.68	36.39	37.74
67	48.89	49.07	58.50	52.15
71	45.67	44.96	46.19	45.61

Jätkub järgmisel lehel

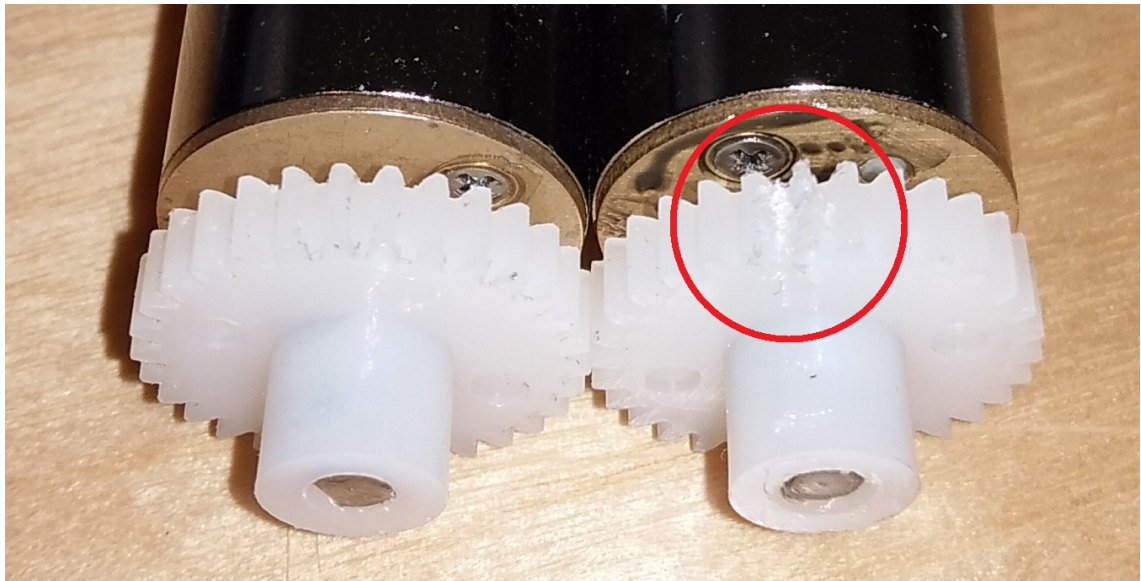
Jätk eelmiselt lehelt

	Keskmine voolutarbimine liikumisel			
Aktuaator	0 → 5 mm	0 → 15 mm	0 → 30 mm	Liikumiste keskmine
72	49.76	48.33	47.52	48.53
73	41.56	40.12	44.12	41.93
74	52.19	52.62	46.35	50.38
75	67.56	66.91	66.10	66.86
73	45.48	43.95	47.81	45.75
77	54.45	53.25	44.69	50.80
81	39.93	38.79	36.80	38.51
86	39.21	36.23	36.89	37.44
91	42.90	41.34	45.99	43.41
92	42.74	43.13	46.49	44.12
93	34.95	36.81	34.55	35.44
94	46.04	47.28	45.80	46.37

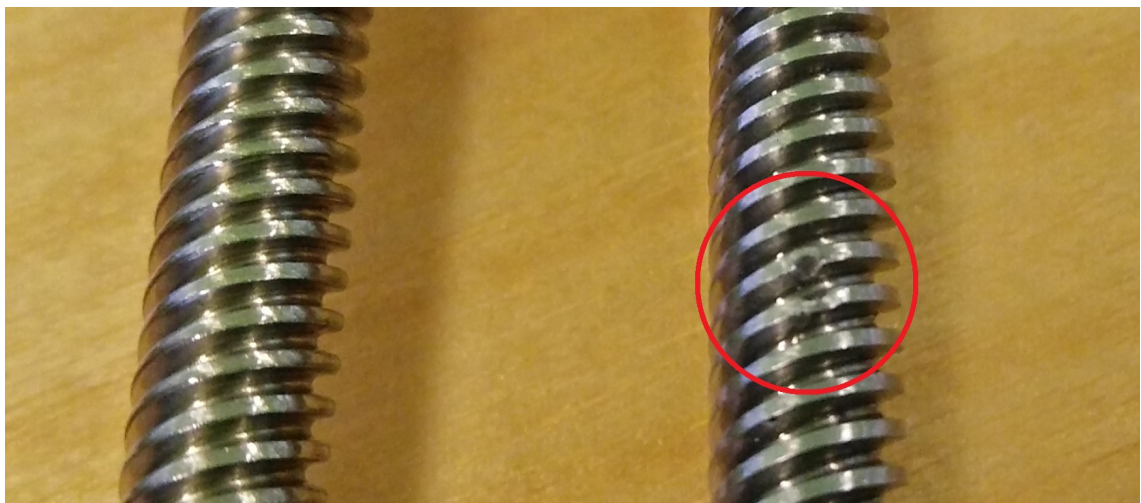


Joonis 8.1: Mannekeeni V25.1 aktuaatorite keskmise voolutarbe histogramm

Lisa 3. Aktuaatori mehaanika vead



Joonis 8.2: Terve hammasratas (vasakul) ja aktuaatori 21 kahjustatud hammasratas (paremal)



Joonis 8.3: Terve keermelatt (vasakul) ja aktuaatori 84 kahjustusega keermelatt (paremal)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Antti Jaaniste

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

**"Robotmannekeenide haldustarkvara ning selle juurde kuuluvate
komponentide parandamine ja edasiarendamine"**

mille juhendajad on Helina Kitsing ja Karl Kruusamäe

- (a) reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - (b) üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile;
 3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **02.05.2017**